Escuela
Politécnica
Superior

Internet of Things: Smart Parking

Grado en Ingeniería en Sonido e Imagen en Telecomunicación

Trabajo Fin de Grado

Autor:
Diego Vigil Peláez

Tutor:
Sergio Antonio Cuenca Asensi

Noviembre 2017

Universitat d'Alacant
Universidad de Alicante

Resumen

En la actualidad, uno de los mayores problemas existentes en una localidad reside en la gestión del sistema de aparcamiento urbano, ya que se pierde mucho tiempo en encontrar una plaza en aquellos lugares más concurridos.

Para ello se propone, mediante el internet de las cosas, la implementación de una red de sensores colocados en cada una de las plazas de estacionamiento de una ciudad. Estos sensores se conectarán al punto wifi más cercano, los cuales transmitirán la información a una base de datos centralizada donde se recogerá el estado de cada una de estas plazas para saber si se encuentran libres u ocupadas, proporcionando esta información a los conductores en tiempo real. Este sistema ayudará al conductor a encontrar una plaza de estacionamiento para su vehículo a través de una aplicación móvil, la cual se conectará a esta base de datos e indicará el estado de los sensores. Estos datos también podrán ser consultados a través de una página web diseñada para tal efecto, así como en diferentes paneles informativos distribuidos por la ciudad.

Para el presente proyecto se utilizarán sensores ultrasónicos HC-SR04 los cuales miden la distancia al obstáculo más cercano por el fenómeno de reflexión del sonido. Los datos recogidos por estos sensores serán transmitidos a una placa Arduino Mega 2560 conectada a una tarjeta ESP32 a través de los puertos serial Rx y Tx, y será esta última la que volcará la información a la nube. Estos sensores están formados por dos transductores: un micrófono y un altavoz. Su funcionamiento, similar al de un SONAR, consiste en emitir un impulso y esperar a que este se refleje en el objeto para ser recogido por el micrófono del dispositivo. En función del tiempo que tarda esta onda en hacer el recorrido y sabiendo que la velocidad del sonido en el aire es de 343 m/s para una temperatura de 20ºC, se puede conocer la distancia entre el sensor y el objeto.

La velocidad del sonido varía ligeramente con la temperatura del orden de 0,6 m/s por cada grado centígrado, además de la humedad y la presión, por lo que este sensor no proporciona medidas muy exactas. Sin embargo, para este proyecto basta con indicar si se detecta o no el objeto. Por tanto, la precisión en la distancia es irrelevante. Independientemente del resultado obtenido, se ha establecido un umbral de 10 cm el cual diferenciará la presencia o no de un vehículo. Otro inconveniente reside en la incompatibilidad de los altavoces y micrófonos frente a los agentes meteorológicos, tales como el agua o el viento, lo que imposibilita su uso en exteriores, en cuyo caso resulta imprescindible modificar el tipo de sensor, considerando los sensores magnéticos como los más utilizados por su robustez y fiabilidad.

El área de implantación de este proyecto será una maqueta de simulación provista de dos paneles solares para mantener un compromiso con el medio ambiente, capaces de suministrar la energía necesaria para el funcionamiento de todo el sistema.

Abstract

In this day and age, one of the greatest issues in a locality resides in the management of the urban parking system, since a lot of time is wasted in finding a parking space in most crowded places.

To improve this, it is proposed, through the Internet of Things, to implement a sensor network placed in each of the parking spaces of a city. These sensors will be connected to the nearest Wi-Fi hotspot, which in turn will be connected to a centralized database where the status of each of these places will be collected to know if they are free or occupied, with the intention of providing this information to the drivers in real time. This system will help the driver find a parking space for his vehicle through a mobile application, which will connect to the database and indicate the status of the sensors. These data can also be consulted through a web page designed for this purpose, as well as in different information panels distributed throughout the city.

For the present project, HC-SR04 ultrasonic sensors will be used, which measure the distance to the nearest obstacle by the phenomenon of sound reflection. The data collected by these sensors will be transmitted to an Arduino Mega 2560 board, which will be connected to an ESP32 board through the serial ports Rx and Tx of both cards, and it will be the latter that will dump the information to the cloud. These sensors are made up of two transducers: a microphone and a speaker. Its operation, similar to that of a SONAR, is to emit a pulse and wait for it to reflect on the object, to be received by the microphone of the device. Depending on the time it takes for this wave to travel and knowing that the speed of sound in the air is 343 m/s for a temperature of 20°C, the distance between the sensor and the object can be known.

The speed of sound varies slightly with temperature in the order of 0.6 m/s for each degree centigrade, in addition to humidity and pressure, so with this sensor very precise measurements cannot be obtained. However, for this project, indicating if the object is detected or not, will be enough. Therefore, accuracy in distance is irrelevant. Regardless of the obtained result, a 10 cm threshold has been established which will differentiate the presence or not of a vehicle. Another disadvantage is the incompatibility of loudspeakers and microphones against weather agents, such as rain or wind, which makes it impossible to use them outdoors. For an external case it is essential to modify the type of sensor, being able to count on magnetic sensors as the most used for its robustness and reliability.

The implementation area of this project will be a simulation mockup in which two solar panels will be installed to maintain a commitment with the environment, capable of supplying the necessary energy for the operation of the entire system.

Por la memoria de mi abuela.

«Sólo hace falta una idea para conseguir el éxito.»

Lori Greiner

Índice de contenidos

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1.- Motivación

En la actualidad, uno de los mayores problemas existentes en una localidad reside en la gestión del sistema de aparcamiento urbano, ya que se pierde mucho tiempo en encontrar una plaza en aquellos lugares más concurridos.

Un estudio **[1]** elaborado por *Siemens* con la colaboración del Ayuntamiento de Madrid arroja que el transporte genera un 41% del total de los gases contaminantes que se producen en una ciudad. De ese porcentaje, los automóviles privados representan el 80% de la contaminación, es decir, 6 toneladas métricas de CO_2 equivalente. El 59% restante corresponde a las emisiones procedentes de edificios e infraestructuras. Según este estudio, es posible obtener una reducción del 10% de las emisiones hasta 2030 introduciendo mejoras en el rendimiento de los automóviles; y entre un 20% y un 30% implantando medidas como coches eléctricos o híbridos, aparcamientos disuasorios o que permitan estacionar en un menor tiempo o la implantación de un sistema de peajes como en Londres.

Según un informe de *Bosch* **[2]**, un 30% del volumen de tráfico del centro de las ciudades es causado por vehículos en busca de un espacio libre donde dejar su vehículo. De media, un usuario pierde 20 minutos cada vez que busca aparcamiento y no sólo eso, sino que el 32% de las multas que se extienden son por estacionamiento incorrecto. Además, sólo el 27% de los edificios residenciales de Madrid tienen parking y el 30% del tráfico y embotellamientos urbanos se atribuye a los usuarios que buscan un sitio donde aparcar su coche.

La declaración final del 16º Congreso Europeo de Aparcamientos (EPA) que tuvo lugar en Dublín **[3]**, concluyó que el aparcamiento tiene un impacto importante en la planificación urbana pero se encuentra infravalorado. El sector del aparcamiento debe prestar más atención al usuario y al hecho de que este sector es un proveedor de servicios. Un tercio de los encuestados cree que el mayor reto del futuro para el sector del parking es gestionar la escasez de plazas y recursos y los crecientes costes de movilidad en áreas urbanas.

Por otra parte, el 65% de las calles de las grandes ciudades españolas superan los 75 dBA recomendados por la Organización Mundial de la Salud (OMS) **[4]**.

La revista económica *The Economist* ha publicado un artículo **[5]** en el que explica en términos económicos el coste que generan los atascos y la búsqueda de parking. Sólo durante el año 2013 el coste de Francia, Inglaterra, Alemania e Italia ha sido de un total de 200 billones de euros, el 0,8% del PIB de estos países. Dos tercios del total de costes son el resultado de combustible malgastado y tiempo que se podría haber utilizado para realizar otras actividades productivas.

1.2.- Objetivos

Para llevar a cabo este proyecto se pretenden alcanzar los siguientes objetivos:

1. Estudio de los sistemas actuales en explotación que abordan el problema de la búsqueda de estacionamiento.
2. Estudio del estado de las tecnologías para el desarrollo de un sistema distribuido de zonas de aparcamiento interconectadas.
3. Desarrollo de un sistema en la nube capaz de gestionar las plazas de garaje y ofrezca información acerca de su estado.
4. Desarrollo de un panel de control en una página web que permita tanto visualizar el estado de las plazas como actuar *in situ*.
5. Desarrollo de una maqueta de simulación autosuficiente que ayude a validar este sistema.
6. Desarrollo de una aplicación móvil que permita consultar las plazas libres en tiempo real.
7. Evaluación del sistema mediante una maqueta y un caso real en un parking público.

Figura 1: Maqueta de simulación

En proyectos futuros, este sistema podría ser implementado en el navegador del coche, el cuál indicaría dónde dirigirse para aparcar, conectándose a esta base de datos y modificando los últimos metros de la ruta dirigiendo el conductor no solo hasta la calle o número que haya indicado al salir, sino hasta el lugar exacto más cercano donde pueda dejar su vehículo en ese preciso instante. Este proceso se haría de forma totalmente automática durante los últimos metros de la ruta.

Capítulo 2

Estado del Arte

2.1.- Introducción a las *Smart Cities*

Una *Smart City* o ciudad inteligente, es una ciudad que aplica las tecnologías de la información y de la comunicación (TIC) con el objetivo de proveerla de una infraestructura que garantice un desarrollo sostenible, un incremento de la calidad de vida de los ciudadanos, una mayor eficacia de los recursos disponibles y una participación ciudadana activa. **[6]** Es decir, las *Smart Cities* son ciudades sostenibles económica, social y medioambientalmente. Desde el punto de vista tecnológico, las *Smart Cities* consisten en interconectar cada elemento de una ciudad para poder controlar su funcionamiento, mejorar su eficiencia y cumplir con los objetivos descritos anteriormente.

2.1.1.- Internet de las Cosas

El Internet de las Cosas, o Internet of Things (IoT, por sus siglas en inglés), un concepto que nació en el Instituto de Tecnología de Massachusetts (MIT), consiste en la interconexión de objetos físicos con internet con el objetivo de ofrecer datos en tiempo real. Cada uno de estos objetos va a disponer de una IP mediante la cual podrá recibir instrucciones o conectarse a un servidor externo y enviar los datos que recoja. Se busca así, obtener información relevante acerca del estado de un objeto o de su entorno para poder realizar determinadas acciones.

«Si una persona se conecta a la red, le cambia la vida. Pero si todas las cosas y objetos se conectan, es el mundo el que cambia.» - Hans Vestberg, CEO de Ericsson.

De este modo, el Internet de las Cosas se trata de la principal tecnología habilitadora de las *Smart Cities*. Se calcula que en 2020, más de 20.000 millones de dispositivos estarán conectados a internet. Cada persona dispondrá de entre dos y seis terminales de IoT. **[7]** El aumento exponencial del número de dispositivos conectados a internet en los últimos años hace que la actual versión 4 del protocolo de Internet (IPv4) que ofrece hasta 2^{32} direcciones de host diferentes, es decir, casi 4.300 millones, no sea suficiente para cubrir la demanda del Internet de las Cosas. Es por esto que Steve Deering y Craig Mudge desarrollaron la versión 6 de este protocolo (IPv6) que admite 2^{128} direcciones diferentes, o lo que es lo mismo, más de 340 sextillones (340.282.366.920.938.463.463.374.607.431.768.211.456). **[8]**

Una vez asignada una dirección única al dispositivo que se pretende monitorizar, se puede extraer información de este dispositivo mediante la tecnología RFID o a través de sensores de diversa índole, como pueden ser sensores de posicionamiento por GPS, niveles de fluidos, gases, temperatura, presión, humedad, proximidad, etc. La tecnología RFID (*Radio Frequency IDentification*, por sus siglas en inglés) o identificación por radiofrecuencia, consiste en identificar mediante un lector sin contacto y a distancia, una tarjeta o etiqueta portada por cualquier persona u objeto. **[9]** Los sistemas RFID se clasifican dependiendo del rango de frecuencias que utilizan en

frecuencia baja (125 ó 134,2 kHz), frecuencia alta a 13,56 MHz, frecuencia ultra alta (UHF entre 868 y 956 MHz) y de microondas a 2,45 GHz.

Esta tecnología se lleva a cabo con etiquetas activas, pasivas o semipasivas, siendo las activas las que requieren de alimentación eléctrica. Las etiquetas semipasivas, semiactivas o asistidas por batería, también disponen de alimentación propia, pero destinada al microchip y no para transmitir la señal. El lector realiza peticiones por radiofrecuencia al chip que integran las etiquetas RFID, las cuales emiten una respuesta permitiendo la identificación del dispositivo en tiempo real. Este código de identificación es único y personalizable. Los *tags* o etiquetas RFID se utilizan en infinidad de aplicaciones como la identificación de objetos en cajas o paléts que se mueven en un almacén, generalmente se utiliza una etiqueta que se adhiere a la superficie del objeto. Las nuevas tarjetas de crédito o débito contactless, llevan incorporadas uno de estos dispositivos que también pueden ser impresos en papel denominados *Smart labels* y que sustituyen así los archiconocidos códigos de barras. Para la identificación de animales, se inserta el tag no tóxico debajo de la piel o en el estómago. Otras aplicaciones con la tecnología RFID pueden ser las llaves de seguridad de un vehículo o las tarjetas de control de acceso a zonas restringidas.

Banda de frecuencias	Descripción	Rango
125 kHz	LF (Baja Frecuencia)	Hasta 50 cm.
13,56 MHz	HF (Alta Frecuencia)	De 8 cm.
868 MHz – 956 MHz	UHF (Ultra Alta Frecuencia)	De 3 a 10 m.
2,45 GHz – 5,4 GHz	Microondas	Más de 10 m.

Tabla 1: Bandas de frecuencia utilizadas en RFID

Por otro lado, esta tecnología se ve complementada por los sistemas de sensorización, los cuales permiten medir luminosidad, presión, humedad, presencia, temperatura y un largo etcétera, ajustándose a las necesidades del Internet de las Cosas.

2.1.2.- Relación entre las *Smart Cities* y el Internet de las Cosas

El modelo ideal de una ciudad inteligente se basa principalmente en los diferentes subsistemas que la compone. Algunos de estos subsistemas son:

- ***Smart Mobility:*** Hace referencia a mejorar la gestión de la movilidad urbana, reduciendo los atascos, causantes del aumento del PIB de entre un 2% y un 4% debido al descenso de la productividad que sufren los trabajadores al llegar tarde a su puesto de trabajo. **[10]**

Los automóviles son los responsables del 38% del consumo energético, además de los principales consumidores de petróleo generando el 25% de los gases de efecto invernadero. Helsinki fue la primera ciudad del mundo en diseñar un sistema de movilidad inteligente conocido como *Mobility-as-a-service* que incluye todos los modelos de transporte para que los ciudadanos puedan seleccionar desde su *smartphone* el tipo de transporte que necesitan en tiempo real. En Londres, la empresa Ford desarrolló un programa piloto que permite alquilar un coche por minutos a través de su aplicación móvil. **[11]**

- ***Smart Government:*** Centralización de la gestión de los servicios públicos que se encuentran a disposición del ciudadano. Un ejemplo de ello es la implementación del nuevo DNI electrónico, el cual permite realizar gestiones a través de internet, agilizando los trámites y reduciendo los tiempos de espera.

- ***Smart Environment:*** Formado a su vez por los siguientes subsistemas: **[12]**

- ***Smart Water:*** Medición de la presión del agua, la temperatura, el nivel, el caudal, el estado y otros datos en la red de captación y suministro de agua para grandes ciudades.

- ***Smart Grid:*** Consumo y eficiencia energética.

- ***Smart Waste:*** Control y sensorización de contenedores de residuos. Consiste en la monitorización de flotas encargadas de la recolección de residuos.

- ***Smart Green:*** Monitorización de la polución, el ruido, el medio natural y perceptual, eco-edificios sostenibles.

Figura 2: Esquema de un *Smart Environment*

Para poder procesar la información de los distintos subsistemas descritos anteriormente, se deben utilizar nodos o puntos de concentración de la información recabada por los sensores que a su vez canalicen los datos a un sistema central capaz de interpretar los resultados y actuar en consecuencia. El conjunto de estos nodos inalámbricos se denomina Red de Sensores Inalámbricos o *Wireless Sensor Network*, por sus siglas en inglés. **[13]**

Esta red de sensores está formada por pequeños dispositivos llamados nodos sensores, alimentados por una batería. Estos dispositivos se encuentran dispersos en redes malladas tipo *ad-hoc* en una determinada zona a monitorizar. Una red *ad-hoc* es aquella en la que no existe un nodo central, sino que todos los nodos ofrecen servicios de encaminamiento para permitir la comunicación de estaciones que no tienen conexión inalámbrica directa.

El modelo seguido por las aplicaciones consiste en realizar una serie de mediciones, digitalizarlas y transmitirlas fuera de la red de sensores a través de un *gateway* o puerta de enlace a una estación base, donde se puede almacenar o procesar la información, antes de acabar en un servidor mayor que permita realizar un histórico o un análisis de los datos recibidos.

Los nodos inalámbricos o motas son dispositivos electrónicos diseñados para formar parte de una red con un objetivo particular. Son capaces de obtener información del entorno, procesarla y transmitirla a su destinatario. El *hardware* de cada uno de estos dispositivos está compuesto por cinco partes claramente diferenciadas.

Figura 3: Arquitectura de un nodo inalámbrico

El procesador se encarga de interpretar y procesar los datos para transmitirlos a otra estación, además de gestionar la memoria. Existen varios productos diferentes en el mercado como pueden ser los microcontroladores, microprocesadores y FPGA.

Microcontroladores: Estos dispositivos incluyen un microprocesador, una memoria y una interfaz para ADCs, UART, SPI, temporizadores y contadores.

Microprocesadores: Se componen de una unidad de control, una unidad aritmético-lógica, varios registros, y en algunos casos, una unidad en coma flotante. Este dispositivo se encarga de ejecutar instrucciones codificadas en números binarios. Algunos microprocesadores de bajo consumo pueden ser: ARM7, Amtel AVR, Intel Xcale, Intel 8051, PIC o el MSP430 de Texas Instruments.

FPGA: *Field Programmable Gate Array*, por sus siglas en inglés, es un circuito integrado diseñado para ser configurado más adelante mediante un lenguaje de descripción de hardware (HDL). Estos dispositivos contienen un array de bloques lógico-programables y una jerarquía de interconexiones reconfigurables que permiten a los bloques conectarse entre sí como muchas puertas lógicas que se pueden interconectar en diferentes configuraciones. Sin embargo, los FPGAs tienen un consumo elevado para este tipo de nodos, por lo que no son una buena opción a tener en cuenta, a menos que se consiga reducir su consumo en los próximos años.

La alimentación puede realizarse mediante baterías o tomas de corriente. Otras técnicas incluyen energías renovables basadas en energía solar, termo generación o energía basada en vibraciones. El consumo de energía viene dado por lo que consumen los sensores, el procesado y la comunicación, siendo esta última la que más energía consume. Los diferentes tipos de batería utilizados, ya sean recargables o no recargables, están clasificadas según el material electroquímico usado para el electrodo como pueden ser NiCd (níquel-cadmio), NiZn (níquel-zinc), Nimh (níquel metal hidruro) y Litio-Ion.

Para la transmisión de los datos recogidos por los sensores se emplea un dispositivo de comunicación inalámbrica denominado transceptor, el cual permite enviar y recibir estos datos vía radio a otros dispositivos que se encuentren dentro de su rango de transmisión. Estos nodos utilizan la banda de transmisión ISM (*Industrial, Scientific and Medical*, por sus siglas en inglés), las cuales son bandas internacionales reservadas para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. No necesitan licencia para su uso y tan sólo es necesario respetar las regulaciones que limitan el nivel de potencia transmitida. El tipo de tecnología utilizada en una comunicación inalámbrica va desde la comunicación óptica mediante láser, infrarrojos y radio frecuencia, siendo esta última la más adecuada con un rango de frecuencias que oscilan entre los 433 MHz y los 2,4 GHz.

Algunos sistemas de comunicación por radio para nodos de redes inalámbricos pueden ser CC3000, CC3100, CC3200, ESP32, ESP8266, XBee, XBee Pro, NRF24L01, entre otros. En cuanto al sistema de comunicación por GSM/GPRS pueden utilizarse otros módulos como el chip SIM900 o el chip A6.

Los sensores son dispositivos *hardware* capaces de medir un cambio en un estado físico, como la presión o la temperatura del entorno en el que se encuentran. La señal analógica detectada se digitaliza con un convertidor analógico digital y se envía a un controlador para ser procesada. Los sensores ideales son pequeños, consumen poco, son autónomos y son capaces de adaptarse al ambiente. Pueden clasificarse en tres categorías:

Sensores activos: Son sensores que sondean el ambiente como pueden ser un radar o un sonar.

Sensores pasivos unidireccionales: Son aquellos que tienen definida la dirección desde donde deben captar la información, como por ejemplo una cámara.

Sensores pasivos omnidireccionales: Son sensores que captan datos genéricos en cualquier dirección como un sensor de presión o temperatura. Son autoalimentados y utilizan la energía para amplificar la señal analógica captada.

Generalmente se utiliza un tipo de memoria desarrollada de la memoria EEPROM denominada memoria flash, la cual permite que múltiples posiciones de memoria sean borradas o escritas en una misma operación mediante impulsos eléctricos, permitiendo su uso a velocidades mayores que otros tipos de memoria. La memoria flash es no volátil, es decir, la información no se pierde en cuanto se desconecta de la corriente. Existen dos categorías de memoria en función de su uso:

- Memoria para almacenar los datos recogidos por la aplicación.
- Memoria para almacenar el programa del dispositivo.

La puerta de enlace o *gateway* encamina la información recibida hasta la estación base. Su objetivo consiste en interconectar una red de sensores con una red de datos (TCP/IP), permitiendo su monitoreo y acceso a la información adquirida por los sensores.

La estación base o controlador de red supervisa la red, controla los paquetes y almacena los datos procesados provenientes de la puerta de enlace. Desde este punto se pueden realizar comunicaciones con otras redes o servicios.

2.2.- Sistemas implementados

2.2.1.- WeGo&Park

WeGo&Park es un sistema de detección inteligente de plazas de aparcamiento. Localiza estacionamientos libres en la vía pública mediante procesado distribuido de vídeo en tiempo real de una red de cámaras estratégicamente colocadas que cubren la zona deseada. Sugiere que este tipo de soluciones son más económicas y versátiles que las basadas en sensores de suelo.

Gracias a este sistema, cualquier usuario puede consultar rápidamente plazas libres para aparcar su vehículo en las proximidades donde están circulando o en el área cercana a su destino, evitando así pérdidas de tiempo, dinero y reduciendo los niveles de contaminación y las emisiones de CO2. El sistema de cámaras se encarga de procesar el vídeo de forma distribuida para detectar la ocupación de plazas de aparcamiento basándose en el tipo de plaza, ya sea gratuita o de pago, además de comprobar la autorización que tiene ese vehículo para estacionar en zonas restringidas o de uso exclusivo.

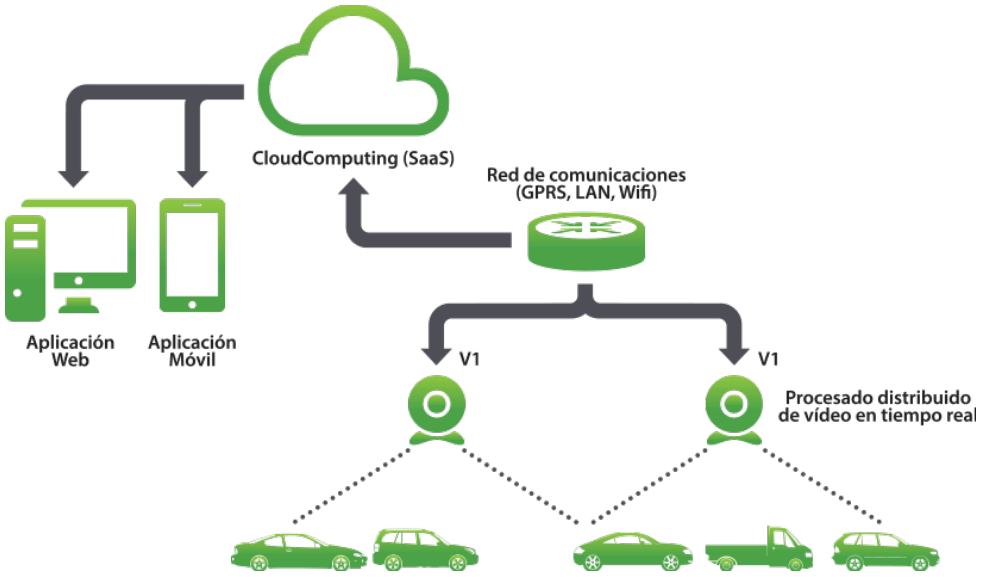


Figura 4: Distribución de la red de *WeGo&Park*

Cualquier usuario puede encontrar y reservar una plaza libre mediante el uso de una aplicación móvil, la cual lo guiará hasta la ubicación detectada. Esta información es también accesible desde su aplicación web.

La empresa que oferta el servicio es Wellness Telecom S.L. con sede en Sevilla, y ha trabajado para otras empresas tales como *Acisa*, *Algesa*, *Alisea*, *Cactus*, *Glece*, la Diputación de Badajoz, *Elecnor*, *Elsamex*, *Endesa Ingeniería*, *Etra*, *Ferrovial*, *Fulton*, *Gamma*, *Gas Natural*, *Germania*, *Humana*, *ISS*, *Movisat*, *Prodetur*, *Promedio*, *Sice* o *Simosa IT* ofreciendo este y otros servicios del internet de las cosas.

Figura 5: Monitorización de las plazas mediante procesamiento de vídeo

2.2.2.- Vatia

La solución que propone esta empresa es un sistema de aparcamientos en zonas azules llamado HEFEST el cual está basado en su plataforma de radio frecuencia WITECH encargada de monitorizar todas las plazas de aparcamiento para su gestión y explotación. Utiliza un sensor ferromagnético pudiendo así controlar la presencia de coches en las plazas de parking con un alto grado de precisión.

El sistema de comunicaciones está basado en nodos de radiofrecuencia que usan banda libre, por lo que no es necesario pagar por la transmisión de datos. Cada uno de estos nodos tiene un alcance de cientos de metros y puede enviar o recibir información a cualquier punto. El nodo central recibirá las lecturas de los otros nodos y volcará esta información al servidor central del sistema.

Figura 6: Parking público ocupado por varios vehículos

El mantenimiento tan sólo requiere comprobar el correcto funcionamiento del emisor y del receptor, reduciendo el coste considerablemente.


Telefónica y Streetline ofrecen una solución de aparcamiento inteligente M2M

Nov 13, 2012, 11:58 ET from Streetline, Inc.

-- Los servicios de aparcamiento inteligentes telemáticos aspiran a reducir las congestiones, mejorando los ingresos y haciendo que sea más sencillo aparcar para los conductores

Figura 7: *Telefónica* y *Streetline* ofrecen una solución M2M

Los administradores de espacios de aparcamiento disponen de una interfaz web desde la cual pueden consultar el estado de cada una de las plazas de manera remota. A partir de la ubicación del conductor, se muestran las plazas disponibles más cercanas a través de la aplicación HEFEST, una vez el usuario haya introducido la dirección de destino deseada. El sistema proporciona las indicaciones necesarias para llegar a la zona seleccionada con *Google Maps*. Además, este sistema se encuentra también integrado con *Google Streetview*.

Los conductores disponen de una aplicación móvil desde la cual pueden consultar la disponibilidad de plazas de aparcamiento en tiempo real y tomar con criterio una decisión sobre a qué zona dirigirse a aparcar.

Este servicio es ofertado por *VATIA Telegestión* ubicada en Málaga y es ofrecido a empresas tales como *ABB*, *Gamesa*, el Ayuntamiento de Málaga, *Areva*, *Cobra*, *Semi*, *OHL*, *Abengoa*, *W2PS*, *ACS*, *Elecnor*, *Ingeteam*, *Siemens*, el metro de Málaga, *Comsa Emte*, *Trainelec*, *Cen*, *FCC*, la Junta de Andalucía, *Jema*, *Alstom*, *International Electronics* o *GreenPower Tech*.

2.2.3.- Streetline

Streetline ha desarrollado un SDK o kit de desarrollo de software, el cual convierte cualquier teléfono móvil en un sensor con la capacidad de proporcionar información sin ningún coste adicional. Este SDK forma parte de su aplicación móvil Parker la cual detecta el movimiento de los vehículos en tiempo real.

Parker es una aplicación móvil que guía por voz a los conductores a las plazas de parking disponibles en tiempo real. Dispone de un sistema de pago para efectuar reservas de aparcamiento, además de un sistema que memoriza la ubicación del vehículo para su posterior localización a pie.

Con la adopción generalizada de los coches interconectados, el posicionamiento global por GPS en el vehículo puede proporcionar su ubicación en tiempo real, facilitando la creación de un mapa con la ocupación de todos los vehículos de una ciudad, conociendo de este modo los espacios

disponibles donde aparcar. El uso de coches como fuente de datos es un sistema muy preciso y crecerá rápidamente en los próximos años.

La plataforma de *Streetline* obtiene datos desde diversas fuentes para recrear un mapa con los espacios disponibles donde poder aparcar en la ciudad. También permite el uso de cámaras de seguridad que simultáneamente pueden utilizarse para comprobar las plazas de aparcamiento libres mediante el tratamiento digital de imágenes.

Streetline es una compañía de propiedad privada con sede central en Foster City, California y que dispone de concesiones de aparcamiento inteligentes en Braunschweig, Alemania y en varias ciudades de Estados Unidos.

2.2.4.- Ecopark

Ecopark es un sistema de monitorización guiado para parking. Utiliza un detector de ocupación para plaza mediante sensores de ultrasonido. Además integra un LED que comunica si está libre u ocupada, así como diferentes sensores que detectan los principales gases emitidos por los vehículos: CO, CO2, NO, NO2 e hidrocarburos. Dispone también de otros sensores que miden los niveles de humedad, temperatura y ruido. Por último, puede incluir una cámara que graba imágenes para un mayor control y seguridad.

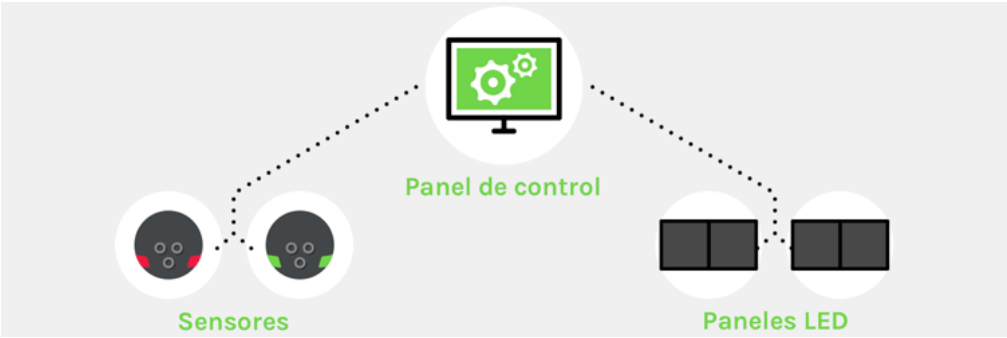


Figura 8: Conexión del sistema *Ecopark*

Todos estos sensores van conectados a un panel de control que permite monitorizar todos los datos y parámetros que se capturan de una forma visual a través de un mapa. Estos datos son recibidos en tiempo real y permite recibir alertas tales como incendios o accidentes.Además disponen de paneles LED distribuidos por todo el aparcamiento que muestra a los usuarios información variable, como la ocupación, señalización, publicidad o avisos relevantes.

Figura 9: Panel informativo con el número de plazas y la temperatura

Ecopark propone soluciones tanto *indoor* como a pie de calle. El conductor será guiado mediante pantallas modulares ofreciendo la cantidad de espacios disponibles tanto a izquierda como a derecha siendo este el último en decidir hacia donde prefiera estacionar su vehículo.

Ecopark es un servicio proporcionado por *i+D3 Equipamientos Tecnológicos S.L.* con sede en Vigo y cuenta con multitud de sistemas tales como un control de alquiler de bicicletas, peajes, asistencia en carretera, control y gestión de accesos y seguridad. Sus servicios de control de estacionamiento y reconocimiento de placas vehiculares han sido implementados en multitud de proyectos, tales como en el sistema de bicicletas de la ciudad de Valencia o en sistemas de guiado de plazas en el parking del Sagrado Corazón de Sevilla o en el parking Marina Miramar de Santa Pola, entre otros.

2.2.5.- IoT Sens

IoT Sens dispone de un sensor magnético que detecta los vehículos estacionados en las plazas de aparcamiento obteniendo así información en tiempo real sobre su disponibilidad. Este dispositivo se sitúa en la superficie de la carretera y dispone de protección anti-vandálica.

Esta empresa utiliza un motor de búsqueda denominado *Elasticsearch* el cual está basado en *Lucene*. Se trata de un sistema multidispositivo con capacidad de búsqueda de texto completo con una interfaz web REST y documentos JSON sin esquema. Además, *Elasticsearch* proporciona un motor distribuido, lo que significa que los índices se pueden dividir en fragmentos y cada fragmento puede tener cero o más réplicas. Cada nodo alberga uno o más fragmentos, y actúa como un coordinador para delegar operaciones hacia el fragmento correcto. Este enrutamiento se realiza de manera automática y está integrado con el *framework* Hadoop. Los desarrolladores pueden escribir trabajos de MapReduce en los datos existentes en el índice en HDFS, lo que permite la búsqueda a través de la API REST y *Elasticsearch*.

Utiliza además el protocolo MQ Telemetry Transport (MQTT), el cual es un protocolo ligero de publicación / suscripción que fluye a través de TCP / IP hacia los sensores remotos y dispositivos de control con un ancho de banda bajo, para entornos poco fiables o comunicaciones intermitentes, haciéndolo ideal para su uso en entornos con limitaciones. MQTT es un estándar OASIS y está diseñado para ser gratuito, simple y fácil de implementar, lo que permite dar servicio a miles de clientes desde un único servidor.

Esta empresa proporciona una interfaz de usuario para tener acceso rápido y fácil a los datos recogidos por su red de sensores. Estos sensores se muestran en un mapa interactivo como un pin con un icono y color diferente en función de su estado. En función del zoom, estos pines se pueden agrupar para evitar la inundación del mapa con demasiados datos. Los usuarios pueden

obtener los datos más relevantes para sus objetivos y reaccionar de acuerdo con el estado del sistema mediante el uso de *widgets* configurables.

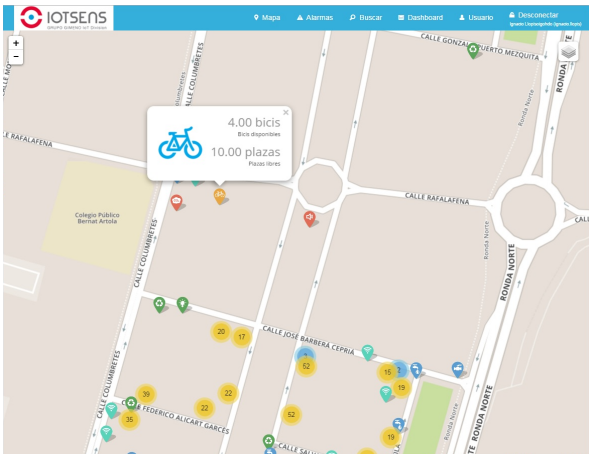


Figura 10: Mapa con las plazas disponibles

Por otro lado, también se proporciona un mecanismo para definir alarmas personalizadas que pueden ser generadas por los sensores o los nodos de comunicación, enviando notificaciones personalizadas a los encargados de reaccionar en cada caso.

IoT Sens del Grupo Gimeno IoT Division es una empresa registrada en Castellón. Actualmente se encuentra desarrollando un proyecto piloto de Smart City, el cual proporciona una visión integral del barrio Pau Gumbau de Castellón y de todos sus servicios, entre los que se encuentra el control de aparcamiento inteligente.

2.2.6.- Urbiótica

La solución inteligente para la gestión del aparcamiento urbano que ofrece esta empresa destaca por guiar a los usuarios evitando las habituales congestiones de tráfico y mejorar la calidad medioambiental, además de ahorrarles tiempo y dinero a los ciudadanos.

El sensor U-Spot detecta la ocupación y la rotación de vehículos en un aparcamiento en tiempo real. Es un sensor autónomo e inalámbrico que utiliza tecnología magnética para detectar la entrada y salida de los vehículos en las plazas de parking además de la duración de su estacionamiento. Puede usarse en parking de pago por tiempo, áreas reservadas a minusválidos, carga y descarga, estacionamientos privados o centros comerciales, pudiéndose integrar con paneles informativos digitales de guiado para la gestión del parking.

Fig. 11: U-Spot

El sensor transmite su información a la red de comunicación utilizando el protocolo propietario U-Sense a bandas de sub-GHz ISM. Esta red canaliza la información del sensor hacia la nube utilizando el protocolo estándar IEEE 802.15.4 a 2,4 GHz. Es apto para cualquier área de estacionamiento exterior y para todo tipo de vehículos. Su rango de temperatura de funcionamiento es de -33ºC a 65ºC y cuenta con un nivel de estanqueidad IP67 y de resistencia mecánica IK10, dándole así inmunidad y robustez ante factores meteorológicos. Se instala en pocos minutos sin cables, se calibra automáticamente y no necesita mantenimiento. Además tiene una vida útil de hasta 12 años y pesa alrededor de 160 gramos.

El enrutador de datos U-Flag recoge los datos que envían los sensores y los retransmite mediante la red Mesh a través de otros U-Flag hasta llegar al U-Box más cercano utilizando frecuencias no licenciadas y un protocolo basado en estándares. Estos enrutadores son capaces de reconectarse a otro U-Box con mejor calidad de señal evitando así la pérdida de conexión entre los sensores y la plataforma. Además, periódicamente generan tramas de datos técnicos que son interpretados por la plataforma y permiten conocer en tiempo real el estado del equipo, detectar incidencias y diagnosticar problemas. Generalmente son instalados en puntos de acceso a la alimentación eléctrica como farolas o fachadas. También disponen de una batería que se carga en cuatro horas y ofrece una autonomía de hasta cuatro días. Otra opción consiste en hacer uso del panel solar U-Sun como fuente de alimentación.

Fig. 12: U-Box

El concentrador de datos U-Box controla la red de sensores inalámbricos y la pasarela que transmite los datos de los sensores hacia la plataforma U-Base mediante Wi-Fi, 3G/GPRS o Ethernet. De aspecto similar a un U-Flag, los U-Box también son colocados en farolas o fachadas con acceso a la alimentación eléctrica. Con unas dimensiones de 7,5 x 20 x 30 cm, ligeramente superiores a un U-Flag, y un alcance máximo de 200 metros, tiene un peso de 1,5 kg.

Finalmente, U-Base se encarga de interconectar los dispositivos distribuidos por el territorio con la nube, procesa y almacena la información y la pone a disposición de aplicaciones y plataformas de consulta en tiempo real. U-Base es una plataforma alojada en la nube que se ofrece como un servicio evitando costes de infraestructura, operación y al mismo tiempo se beneficia de la escalabilidad de las tecnologías *Cloud Computing*. Dispone de una API basada en servicios web SOAP y REST, así como tecnologías Pub/Sub, que facilita la integración con otros sistemas tanto para la transferencia de información, como para la configuración de los parámetros del sistema.

U-Admin es uno de los componentes tangibles del U-Base a través del cual se pueden administrar y monitorizar todos los elementos del sistema. Se trata de una aplicación alojada en la nube junto con U-Base que se despliega en la Plataforma Cloud de *Urbiótica*.

Figura 13: Panel de control de *Urbiótica*

Esta aplicación se compone de varias herramientas que permiten configurar los proyectos, activar y dar de alta los dispositivos mediante un código QR y monitorizarlos, visualizar la información que se genera y administrar los usuarios de los servicios web del U-Base pudiendo asignar diferentes niveles de acceso y gestión en función de los perfiles que se definan. Existe una versión para *smartphones* que puede usar el instalador para agilizar el proceso de instalación y mantenimiento.

Esta empresa basa su propuesta en las soluciones que satisfacen las necesidades de las administraciones y los ciudadanos. Para ello, se desarrolla un sistema *end-to-end* que provee de una red de sensores magnéticos U-Spot, los cuales hacen llegar los datos recogidos a la Plataforma en la Nube de *Urbiótica*, desde la cual se pueden gestionar mediante la aplicación U-Admin. Los usuarios tendrán que utilizar la aplicación U-Base para encontrar el estacionamiento libre más cercano o hacer uso de los paneles informativos instalados para tal efecto.

Urbiótica es una empresa registrada en Barcelona y han trabajado para la monitorización del tráfico urbano en Sète (Francia), además de proporcionar estacionamiento inteligente en Ibiza, Castellón, L'Escala, Las Condes (Chile) o en el parking de la fábrica de Audi en Ingolstadt (Alemania).

Un sistema de monitorización para 5.000 plazas de parking en Alemania

En total 22 sensores emiten información sobre el número de plazas disponibles, optimizando el espacio y reduciendo la circulación en el aparcamiento.

A través de 22 sensores de aforo de parking inalámbricos colocados en un aparcamiento con más de 5.000 plazas, es posible detectar los vehículos que entran y salen de cada sector y transmitir información sobre el número de plazas disponibles con las que cuentan los conductores que quieran dejar allí su vehículo.

Es el proyecto que ha llevado a cabo Urbiotica en el parking de la fábrica de Audi en Ingolstadt (Alemania) y con el que se optimiza el uso de los espacios y disminuye la circulación generada por los coches en busca de sitios libres, algo que repercute en una mejor experiencia para los trabajadores de la factoría, que ya no se ven obligados a perder tiempo buscando un espacio.

Figura 14: Extracto de la noticia de un proyecto de *Urbiótica* en Alemania

2.2.7.- Libelium

Libelium utiliza sensores integrados en la plataforma *Waspmote*, la cual permite conectar cualquier sensor a cualquier plataforma en la nube mediante cualquier tecnología inalámbrica disponible.

Esta empresa utiliza sensores magnéticos circulares con un diámetro de 23 cm. Sus baterías de litio tienen una vida útil de entre 4 y 6 años y se colocan en la superficie del asfalto. La temperatura de funcionamiento es de entre -20ºC y 65ºC y cuenta con un nivel de estanqueidad IP67.

Figura 15: Sensor magnético de *Libelium*

Este dispositivo es compatible con las tecnologías *Sigfox* y *LoRaWAN* tanto para Europa como para EEUU.

Plug&Sense! Smart Parking dispone de dos modos de funcionamiento, uno para el día y otro para la noche. Este último ha sido desarrollado para utilizarlo cuando existen menos probabilidades de que un vehículo ocupe o abandone una plaza de estacionamiento, como por ejemplo, durante la noche. De esta forma se consigue una economización de la batería y una reducción de datos redundantes transmitidos.

Figura 16: Router *Meshlium*

Meshlium es un router Linux que funciona como puerta de enlace de las redes de sensores *Waspmote*. Puede manejar hasta seis interfaces inalámbricas distintas: 2,4 GHz Wi-Fi, 5 GHz Wi-Fi, 3G/GPRS, Bluetooth, XBee y LoRa. Además, también puede integrar un módulo GPS para aplicaciones para móviles y vehículos y puede alimentarse mediante una batería o una placa solar. Con estas características, además de poseer una carcasa de aluminio con protección IP65, el router *Meslium* puede ser colocado en cualquier lugar al aire libre.

El nodo de sensorización *Waspmote Plug&Sense!* es el encargado de recoger todos los datos de los sensores y enviarlos al router *Meshlium* descrito anteriormente, el cual a su vez vuelca todos esos datos en la nube.

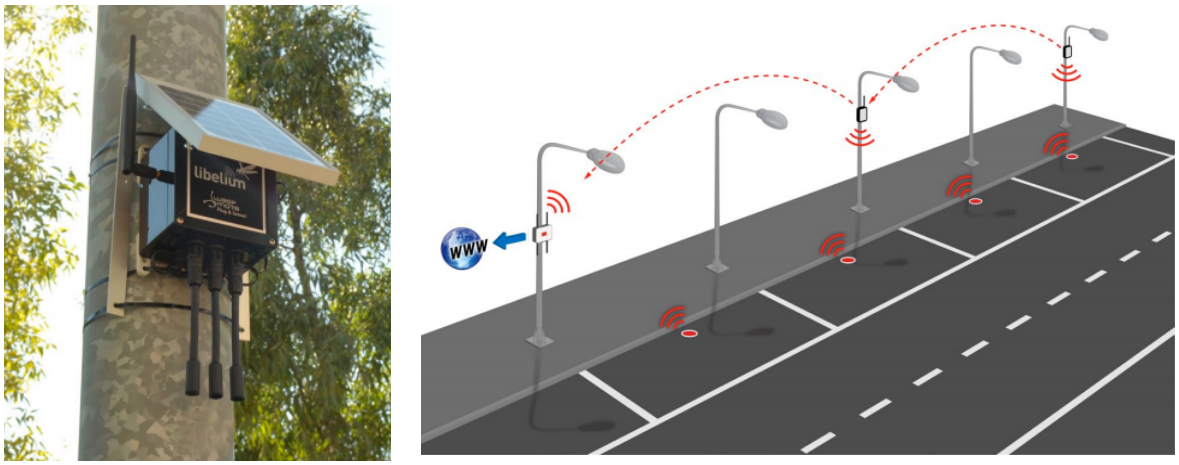


Figura 17: Router instalado en una farola (a la izquierda)
y comunicación entre dispositivos (a la derecha)

Un ejemplo pionero de esta tecnología es el proyecto Smart Santander, el cual ha sido desarrollado por varias compañías e instituciones entre las que se incluyen Telefónica I+D y la Universidad de Cantabria, tiene como objetivo diseñar, desplegar y validar en Santander y su entorno una plataforma compuesta de sensores, cámaras y pantallas para ofrecer información útil a los ciudadanos. Para ello se han implementado 375 *Waspmotes* con el objetivo de controlar los aparcamientos de la ciudad.

La ciudad de Santander fue dividida en 22 zonas diferentes. Cada una de ellas con un *Meshlium* para recoger los datos de los sensores. Cada zona fue configurada con diferentes parámetros de red de tal forma que se crearon redes independientes las cuales trabajan en canales de frecuencia distintos evitando así posibles interferencias entre ellos.

Los 375 sensores magnéticos están conectados a su *Waspmote* más cercano y envían los datos recogidos periódicamente a los repetidores y estos al router Meshlium correspondiente, el cual almacena los datos y actualiza la información disponible para los ciudadanos.

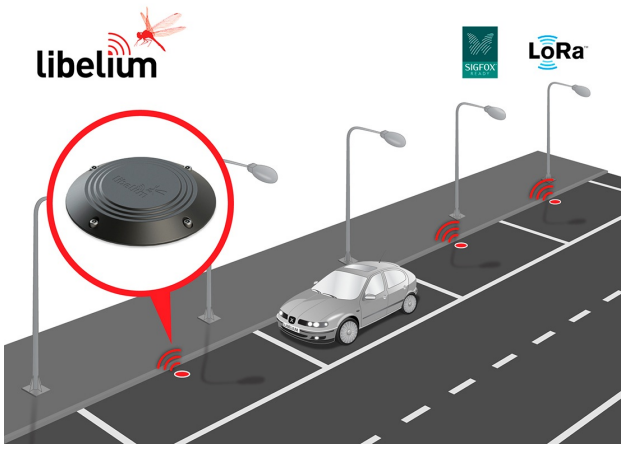


Figura 18: Sensor magnético instalado en cada plaza de estacionamiento

Además, se han colocado una serie de paneles en la ciudad que indican el número de plazas de aparcamiento libres. Esta información se actualiza cada cinco minutos permitiendo así a los ciudadanos encontrar una plaza libre donde estacionar su vehículo en el menor tiempo posible.

Figura 19: Panel informativo de la ciudad de Santander

El estado de estas plazas de aparcamiento también se encuentra disponible en un mapa interactivo que puede ser consultado en internet por los ciudadanos en cualquier momento.

Figura 20: Mapa de Santander con el estado de las plazas en tiempo real

Este proyecto se trata de una red a gran escala que permite a los investigadores de todo el mundo probar diferentes algoritmos en un entorno real.

Libelium Comunicaciones Distribuidas S.L es una empresa con sede en Zaragoza.

2.2.8.- WazyPark

La solución que propone esta empresa carece de una red de sensores, por lo que recurre a la colaboración ciudadana para que, mediante su aplicación para móviles, notifiquen la existencia de huecos libres cada vez que abandonen el estacionamiento, dejándolo disponible para los demás usuarios. Todos los usuarios deben descargarse una aplicación móvil mediante la cual se notifican las plazas libres más cercanas con la colaboración ciudadana. Para su funcionamiento, esta aplicación requiere del factor humano y de la fiabilidad y buena fe de sus usuarios, y sincroniza toda la información en tiempo real a través de internet.

Una nueva actualización permite vincular la aplicación con el *bluetooth* del coche, conociendo así cuándo se arranca el vehículo. De esta forma, la plaza libre será compartida automáticamente con el resto de la comunidad.

Wazypark ofrece una serie de recompensas al ayudar a otros usuarios a estacionar su vehículo. De este modo se consiguen dos puntos al avisar si se deja una zona de aparcamiento, tres puntos si se avisa con antelación y un punto si la zona sirve a otro usuario o al comunicar que esa zona ya no se encuentra disponible. Cada cinco puntos se obtendrá un euro de descuento en gasolineras *Repsol*. Para ello se genera un código QR que habrá que mostrar al momento de efectuar el pago. Otras empresas como *Jazztel*, *Just Eat* o *Hailo* también forman parte del sistema de puntos. Por otro lado, esta aplicación ofrece información relevante como el precio del combustible en las diferentes gasolineras o la ubicación y el precio de los diferentes parkings privados de la zona.

WazyPark S.L. es una empresa española con sede en Madrid.

Capítulo 3

Estudio de la tecnología

3.1.- Diseño físico

3.1.1.- Capa de enlace

a) IEEE 802.11

El protocolo de comunicaciones IEEE 802.11 **[14]** define el uso de los dos niveles inferiores del modelo OSI (capa física y de enlace) y especifica las normas de funcionamiento de una red de área local inalámbrica (WLAN). **[15]**

WiFi es una marca de la Alianza WiFi, una organización comercial que adopta, prueba y certifica que los equipos cumplen con los estándares 802.11 relacionados a redes inalámbricas de área local. El comité IEEE 802 que estandariza las redes de área local fue creado en febrero de 1980, de ahí su nombre (año 80, segundo mes). **[16]**

Existen multitud de estándares de esta norma, pero los más conocidos internacionalmente son los estándares IEEE 802.11b **[17]**, IEEE 802.11g **[18]** e IEEE 802.11n **[19]**, ya que utilizan la banda de 2,4 GHz disponible en la mayoría de los países, con velocidades de 11, 54 y 300 Mbps respectivamente.

Estándares IEEE	IEEE 802.11b	IEEE 802.11g	IEEE 802.11n	IEEE 802.11ac
Año de lanzamiento	1999	2003	2009	2013
Frecuencia	2,4 GHz	2,4 GHz	2,4 y 5,4 GHz	5,4 GHz
Velocidad teórica	11 Mbps	54 Mbps	600 Mbps	6,93 Gbps
Velocidad práctica	6 Mbps	22 Mbps	100 Mbps	100 Mbps
Ancho de banda	22 MHz	20 MHz	20-40 MHz	80-160 MHz
Modulación	DSSS	OFDM	OFDM mejorado	256-QAM
Alcance	460 m	460 m	820 m	300 m

Tabla 2: Comparativa de estándares IEEE 802.11

Actualmente se maneja el estándar IEEE 802.11ac **[20]** conocido como WiFi 5G ó WiFi Gigabit, el cual opera en la banda de los 5 GHz, habilitada en enero de 2014, pudiendo disfrutar de un menor número de interferencias, aunque con un 10% menos de alcance, debido a que a mayores frecuencias, el alcance se reduce.

En el CES 2016 (*Consumer Electronic Show*, Feria Electrónica de Consumo, por sus siglas en inglés), dieron a conocer el estándar IEEE 802.11ad **[21]** que operaría en la banda de los 60 GHz con tasas de transferencia de hasta 6 Gbps prácticos. Ese mismo año se implementó el estándar

IEEE 802.11ah **[22]** que consigue un alcance de hasta 1000 metros reduciendo la frecuencia en la que opera a 0,9 GHz.

Desde el estándar IEEE 802.11ac se implementó la tecnología *Beamforming*, derivada de la tecnología MIMO (*Multiple-Imput Multiple-Output*), una tecnología que consigue "deformar" la onda para que esta salve obstáculos tales como muebles o paredes, de tal forma que siempre encuentre el camino más "libre" hasta el punto receptor. **[23]**

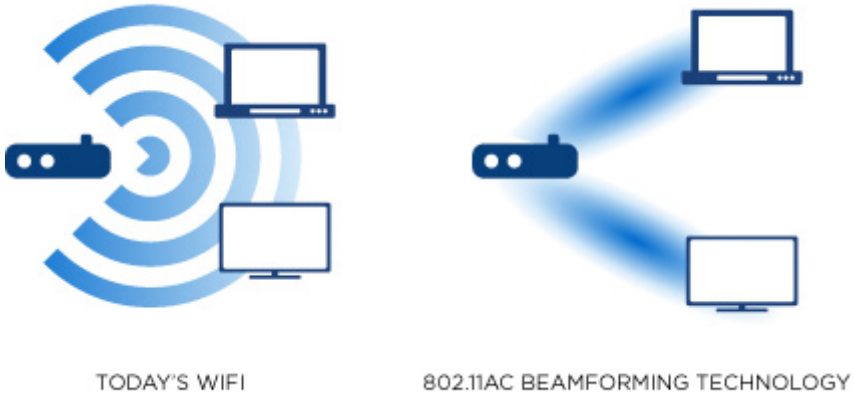


Figura 21: Comparativa entre el WiFi actual y el nuevo sistema *Beamforming*

Existen varias alternativas que garantizan la seguridad de estas redes, siendo las más comunes los protocolos WEP, WPA o WPA2. **[24]**

WEP (*Wired Equivalent Privacy* o Privacidad Equivalente a una red Cableada) es un protocolo de red muy poco seguro. Ofrece un cifrado de nivel 2 basado en el algoritmo RC4 que utiliza claves de 64 ó 128 bits.

WPA (*WiFi Protected Access* o Acceso WiFi Protegido) hace uso de un vector de inicialización de 48 bits y una clave de cifrado de 128 bits. Utiliza el Protocolo de Integridad de Clave Temporal (TKIP, *Temporal Key Integrity Protocol*, por sus siglas en inglés), el cual le permite usar una clave de cifrado diferente para cada paquete de la red, a diferencia del cifrado WEP que utiliza la misma clave para todos los paquetes.

WPA2 se trata del estándar IEEE 802.11i, el cual dispone de cifrado AES **[25]** (*Advanced Encryption Standard* o Estándar de Encriptación Avanzada, por sus siglas en inglés), que utiliza el algoritmo *Rijndael* particularizado para un bloque fijo de 128 bits y tamaños de llave de 128, 192 y 256 bits. AES opera en una matriz de 4x4 bytes llamada state.

Los dispositivos que forman parte de esta tecnología son:

Puntos de acceso: Son dispositivos que generan una red WiFi a la que pueden conectarse otros dispositivos de manera inalámbrica. Se puede conseguir una mayor cobertura añadiendo más puntos de acceso o conectando antenas más grandes que amplifiquen la señal.

Repetidores: Son equipos que extienden la cobertura de una red creando una señal más limpia y potente. Algunos pueden funcionar también como puntos de acceso.

Routers: Son dispositivos compuestos por un enrutador encargado de interconectar redes, un punto de acceso, y generalmente un conmutador que permite conectar otros equipos por cable (USB o Ethernet). Normalmente se encargan de conectar una red local con internet.

WiFi Direct es una norma que permite conectar entre sí varios dispositivos WiFi sin la necesidad de un punto de acceso intermedio. En su lugar, WiFi Direct incrusta un Punto de Acceso en forma de software (*Soft AP*, *Software Access Point*) proporcionando seguridad WPA2. La conexión puede completarse al accionar un botón o mediante un código PIN.

b) IEEE 802.3

El protocolo de comunicaciones IEEE 802.3 **[26]** define el estándar de las redes basadas en Ethernet. Este protocolo comprueba si alguien está usando el medio antes de transmitir y detecta las colisiones desde el emisor mediante el algoritmo CSMA/CD **[27]** (*Carrier Sense Multiple Access with Collision Detection*, o Acceso Múltiple con Escucha de Portadora y Detección de Colisiones, por sus siglas en inglés).

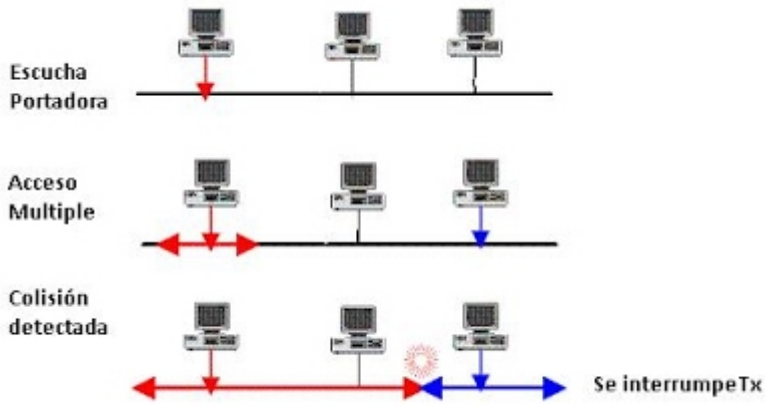


Figura 22: Una estación debe esperar antes de transmitir porque el canal se encuentra ocupado

En 1970, Robert Metcalfe escribió un artículo describiendo un protocolo que mejorase sustancialmente el rendimiento de la red ALOHA **[28]**, la cual desperdiciaba más del 80% del ancho de banda debido a estaciones que intentaban emitir al mismo tiempo. La red ALOHA fue un sistema de redes de computadoras que Norman Abramson estaba desarrollando en Hawái en la década de los 70.

La red Ethernet **[29]** define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI. Su nombre hace referencia a la teoría física ya abandonada según la cual las ondas electromagnéticas viajan por un fluido denominado éter que supuestamente llenaba todo el espacio.

Nombre	Ethernet	Año	Cable	Longitud máxima del segmento	Nodos/ segmento	Topología
10Base5	802.3	1983	Coaxial grueso	500 m	100	Bus
10Base2	802.3a	1985	Coaxial fino	185 m	30	Bus
10BaseT	802.3i	1990	Par trenzado	100 m	1024	Estrella
10BaseF	802.3j	1993	Fibra óptica	1000 m	1024	Estrella

Tabla 3: Principales variantes de Ethernet a 10 Mbps

Nombre	Ethernet	Año	Cable	Longitud máxima del segmento	Topología
100BaseTX	802.3.u	1995	Par trenzado UTP Cat5 ó STP	100 m	Estrella
100BaseT4	802.3.u	1995	Par trenzado UTP Cat3	100 m	Estrella
100BaseFX	802.3.u	1995	Fibra óptica	412 m (half-dúplex) 2000 m (full-duplex)	Estrella
100BaseT2	802.3y	1998	Par trenzado UTP	100 m	Estrella

Tabla 4: Principales variantes de Fast Ethernet a 100 Mbps **[30]**

100BaseT4, a diferencia de 100BaseTX, separa la topología física de la lógica. Además, en 100BaseT4 se utiliza en todo momento tres pares para cada dirección de datos. **[31]**

Nombre	Ethernet	Año	Cable	Longitud máxima del segmento	Topología
1000BaseLX	802.3.z	1998	Fibra óptica monomodo Fibra óptica multimodo	5000 m 550 m (50 µm) 440 m (62,5 µm)	Estrella
1000BaseSX	802.3z	1998	Fibra óptica multimodo	220 m (50 µm) 500 m (62,5 µm)	Estrella
1000BaseCX	802.3z	1998	Cable biaxial STP	25 m	Estrella
1000BaseT	802.3ab	1999	Par trenzado UTP Cat5e	100 m	Estrella

Tabla 5: Principales variantes de Gigabit Ethernet a 1000 Mbps **[32] [33] [34] [35] [36] [37]**

Nombre	Ethernet	Año	Cable	Longitud máxima del segmento	Topología
10GBaseSR	802.3ae	2002	Fibra óptica multimodo	300 m (50 µm) 26-82 m (62,5 µm)	Estrella
10GBaseLR	802.3ae	2002	Fibra óptica monomodo Fibra óptica multimodo	10 km 260 m (50 µm) 220 m (62,5 µm)	Estrella
10GBaseLX4	802.3ae	2002	Fibra óptica monomodo Fibra óptica multimodo	10 km 300 m (50 µm) 300 m (62,5 µm)	Estrella
10GBaseER	802.3.ae	2002	Fibra óptica monomodo	40 km	Estrella
10GBaseT	802.3an	2006	Par trenzado UTP Cat 6 Par trenzado STP Cat 6	55 m 100 m	Estrella
10GBaseCX4	802.3ak 802.3ap	2004 2007	Cable biaxial Par trenzado UTP Cat 5e	15 m 15 m	Estrella

Tabla 6: Principales variantes de Ethernet a 10 Gbps **[38] [39]**

Actualmente existen velocidades mayores como las desarrolladas a partir de 2010 que pueden alcanzar los 40 Gbps según el estándar 802.3ba para una distancia máxima de 10 metros con cables de pares trenzados UTP de categoría 7 ó distancias de entre 100 y 125 metros para fibra óptica multimodo hasta los 10 km con fibra óptica monomodo. También se están trabajando en velocidades que incluso llegan hasta los 400 Gbps como por ejemplo las que se definen en el estándar 802.3bs **[40]**.

c) IEEE 802.15

El protocolo de comunicaciones 802.15 **[41]** se define como un grupo de trabajo dentro de IEEE 802 especializado en redes WPAN (*Wireless Personal Area Networks* o Redes Inalámbricas de Área Personal, por sus siglas en inglés). Este protocolo se divide en 10 subgrupos que van desde el 802.15.1 hasta el 802.15.10.

IEEE 802.15.1 desarrolla un estándar basado en la especificación 1.1 de Bluetooth, tecnología que se encuentra disponible en el módulo ESP32. *Véase apartado 4.4.2: Módulo ESP32.*

IEEE 802.15.4 en su epígrafe f hace referencia a las redes RFID definidas en el apartado *2.1.1: Internet de las Cosas.*

Ninguna de estas tecnologías será implementada en este proyecto.

d) IEEE 802.16

WiMAX (*Worldwide Interoperability for Microwave Access*, o Interoperabilidad Mundial para Acceso por Microondas, por sus siglas en inglés), **[42]** es un estándar recogido en la norma 802.16, una especificación para las redes de acceso metropolitanas inalámbricas de banda ancha fijas publicada el 8 de abril de 2002. El Forum WiMAX es el encargado de certificar la interoperabilidad de los productos basados en este estándar de comunicaciones. Finalmente, esta tecnología no consiguió imponerse al LTE.

e) Redes móviles **[43] [44]**

La tecnología móvil comenzó en 1973 cuando Martin Cooper hizo su primera llamada. A principios de los 80 se empezaron a desplegar redes de telefonía móvil en los países desarrollados. Esto es lo que se considera telefonía móvil de primera generación.

El primer estándar de telefonía móvil totalmente digital creado por 3GPP fue GSM. Otras versiones mejoradas son GPRS o EDGE, también conocido como EGPRS o GPRS mejorado. Además, existen otras tecnologías como IS95 o D-AMPS que forman parte del proyecto 3GPP2. Todas ellas pertenecen a la segunda generación móvil o 2G. El GSM se creó para las llamadas de voz, algo insuficiente para poder navegar por internet, así que la ITU (Unión Internacional de Telecomunicaciones, por sus siglas en inglés), creó a finales de los 90 el comité IMT-2000 para desarrollar la tercera generación o 3G con una velocidad mínima de transferencia de datos de 2 Mbps. Nació así el UMTS, una evolución de GSM. Poco después estas normas se iban mejorando con otras tecnologías como HSPA considerada como 3,5G.

Nombre	Release	Velocidad descarga	Velocidad subida
HSDPA	5	14,4 Mbps	384 Kbps
HSUPA	6	14,4 Mbps	5,76 Mbps
HSPA+	7	28 Mbps	11,5 Mbps
HSPA+ MIMO	8	42 Mbps	11,5 Mbps

Tabla 7: Comparativa de velocidades de HSPA

En 2005, 3GPP, el mismo organismo que creó los estándares GSM y UMTS, creó el estándar LTE para unificar los distintos estándares 3G y conseguir incluso mejorar sus velocidades. Esto podría ser debido a su competidor WiMAX que ya ofrecía velocidades superiores a UMTS. De este modo, el estándar LTE (*Long Term Evolution*, por sus siglas en inglés) consigue velocidades más elevadas a costa de utilizar un mayor espectro y mejorar la eficiencia espectral. Sin embargo, LTE seguía sin ser 4G ya que no cumplía con los requisitos necesarios.

En 2008, la ITU volvió a crear otro comité, esta vez denominado IMT-Advanced para definir lo que se debía considerar como tecnología de cuarta generación o 4G, estableciendo entre otras cosas, velocidades de transmisión de 100 Mbps para movilidad alta y 1 Gbps para movilidad baja. Fue entonces cuando se creó la tecnología LTE-Advanced que no se comercializaría hasta principios de 2012.

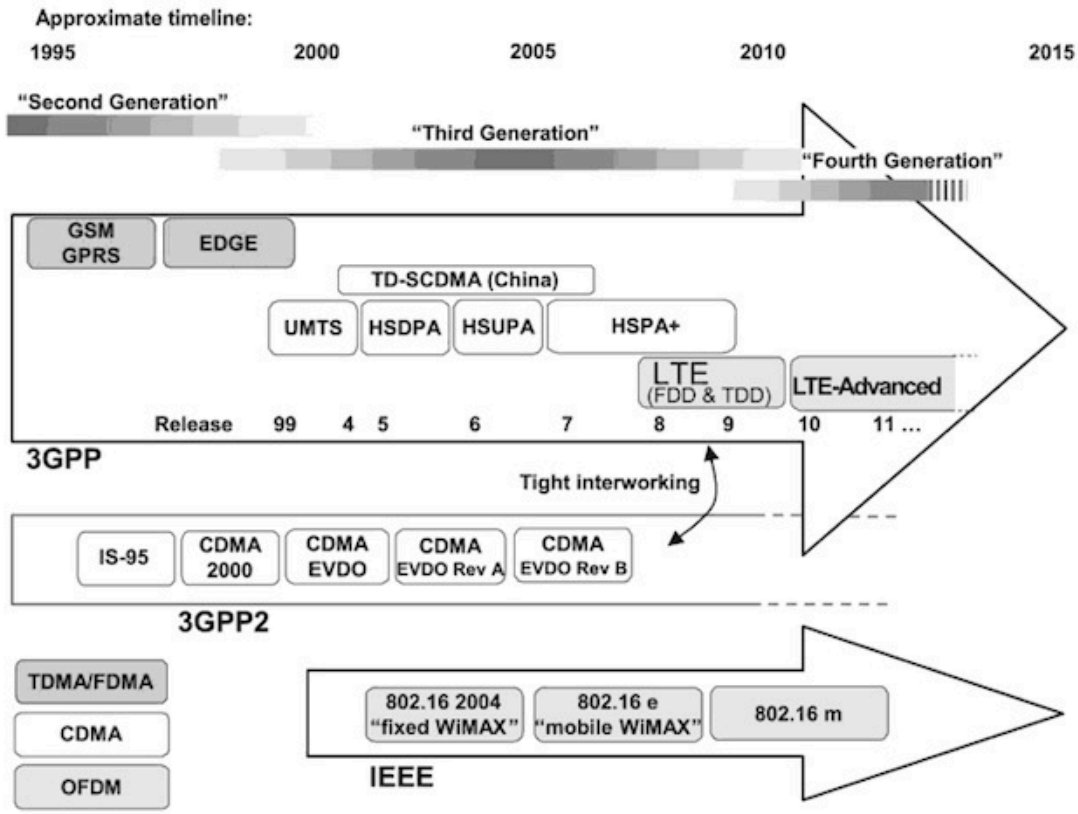


Figura 23: Comparativa de las diferentes redes móviles

Actualmente, se encuentran desarrollando las redes de quinta generación o 5G **[45]** con velocidades que superan los 7 Gbps en movilidad baja, y 1,2 Gbps en movilidades de 110 km/h. Aunque las actuales redes 4G utilizan frecuencias que van entre los 800 MHz y los 2,6 GHz, la nueva red 5G usará frecuencias más elevadas, lo que reducirá su alcance. Sin embargo, también se reducirá la latencia hasta valores cercanos a un milisegundo y la vida de las baterías se incrementará en hasta un 10%. Se espera que para el 2020 esta red comience a desplegarse. Antes necesitarán un par de años para la creación del estándar, y uno más para el desarrollo de productos compatibles.

3.1.2.- Capa de red

Esta capa opera mediante el protocolo de internet (IP, *Internet Protocol*, por sus siglas en inglés) en sus versiones 4 ó 6. Es la encargada de enviar los datos desde el origen hasta el destino. Para ello, los datagramas contienen las direcciones IP de ambos extremos y van circulando a través

de nodos de conmutación haciendo uso de los diferentes protocolos de encaminamiento (RIP, OSPF, IGRP, EGP, BGP) tal y como haya sido configurado por el administrador de la red.

En 1973 se diseñó la versión 1 de TCP, documentada en RFC 675. La versión 2 data de marzo de 1977. En agosto de ese mismo año, Jon Postel se dio cuenta de que este protocolo no iba en la dirección correcta al violar el principio de estratificación ya que se trataba de un protocolo a nivel de host de extremo a extremo y como protocolo de empaquetado y encaminamiento de internet. Por tanto, decidieron dividir los protocolos TCP e IP creando la versión 3 en la primavera de 1978. Finalmente se agregó una estabilidad en su cuarta versión, que es la que todos conocemos como IPv4 **[46]**.

IPv4 fue creado para soportar la entrega eficiente de flujos de paquetes a destinos únicos o múltiples, requiriendo tasas de datos garantizadas y retardo controlado. En otras palabras, estaba tratando de resolver problemas de calidad de servicio desde el Protocolo de Internet original. Con IPv5 trataron de encontrar una forma de transmitir voz a través de redes de conmutación de paquetes, pero esta versión fallida que intentaba resolver este y otros problemas, no tuvo el éxito que se buscaba, ya que las mejoras de red de los últimos años tales como el ancho de banda, las aplicaciones o la compresión fueron suficientes para continuar con la versión 4 de este protocolo. Sin embargo, la escasez de direcciones entre otros motivos por el internet de las cosas, hizo imprescindible dar el paso a IPv6 la cual admite hasta 2^{128} direcciones diferentes.

3.1.3.- Capa de transporte

La capa de transporte hace referencia a los protocolos que permiten una transferencia de segmentos de extremo a extremo. Esto se lleva a cabo mediante los protocolos TCP o UDP. Esta capa permite corregir errores, y controla el flujo, la congestión y la segmentación de los datos.

El protocolo de control de transmisión (TCP, *Transmission Control Protocol*, por sus siglas en inglés), es uno de los más utilizados en internet. Este protocolo da soporte a la mayoría de las aplicaciones de la red, tales como navegadores, correo electrónico o transferencia de ficheros. Fue creado a principios de los 70 por Vint Cerf y Robert Kahn **[47]**. Se trata de un protocolo con conexión y estado, es decir, garantiza una transmisión de los paquetes en orden. Si un paquete no llega a su destino o llega corrupto, el receptor solicita una nueva transmisión, descarta todos los paquetes que llegaron posteriores a este y el emisor volverá a transmitirlos todos desde ese punto.

Para determinadas aplicaciones en las que se requiera una transmisión más inmediata sin importar la pérdida de alguno de los paquetes, se puede utilizar el protocolo sin conexión UDP (*User Datagram Protocol*, por sus siglas en inglés). Al no existir una sesión activa entre el emisor y el receptor, este protocolo no garantiza la entrega de los paquetes, ni su orden de llegada.

3.1.4.- Capa de aplicación

Esta es la séptima capa del modelo OSI y el cuarto de la pila TCP. Es la encargada de interactuar entre el usuario y la capa 6 (de presentación) para que realice el envío del paquete.

Protocolos:

- FTP: Transferencia de ficheros.
- FTPS: FTP con capa de seguridad.
- TFTP: FTP por UDP. Para pequeños archivos.
- DNS: Sistema de nombres de dominio.
- DHCP: Configuración dinámica de anfitrión.
- HTTP: Transferencia de hipertexto. Acceso a páginas web.
- HTTPS: HTTP con capa de seguridad.
- SMTP: Envío de correo electrónico.
- POP: Recepción de correo electrónico.
- TELNET: Acceso a equipos remotos.
- SSH: Shell segura. Telnet encriptado.
- LDAP: Acceso a directorios.
- XMPP: Intercambio de XML. Mensajería instantánea.

Servicios:

- Aplicaciones de red.
- WWW: World Wide Web.

3.2.- Diseño lógico

3.2.1.- Bloques funcionales

Los bloques funcionales de un sistema del internet de las cosas son seis:

- **Aplicación:** Proporciona una interfaz gráfica que permite monitorear el sistema e interactuar con los usuarios. En este proyecto se pretende desarrollar una aplicación web y una aplicación móvil. La primera permite monitorear e interactuar, mientras que la segunda sólo permite monitorear.
- **Servicios:** Son las prestaciones de las aplicaciones. Para este proyecto, el servicio fundamental es el de indicar el estado de varias plazas de estacionamiento. Otros servicios secundarios son: el monitoreo de los contenedores de reciclaje, el envío de un email cuando estos se encuentran llenos, el control de los semáforos o el control del alumbrado.
- **Comunicación:** Este bloque enlaza los dispositivos entre sí permitiendo el envío de la información. En este proyecto se utilizarán dos ESP32 descritos en el apartado 4.4.2.
- **Dispositivo:** Se trata del sensor en cuestión. Para este proyecto se utilizarán principalmente 26 sensores HC-SR04 descritos en el apartado tal.

- **Control y seguridad:** Estos bloques garantizan la autenticidad, integridad y autorización requeridos. Para ello, la plataforma *Carriots* dispone de claves o *apikeys* que definen privilegios y visibilidad. Además, su conexión es mediante el protocolo HTTPS el cual permite cifrar conversaciones con la API REST. Permite cifrar el código de los eventos así como también hace uso del *hash* HMAC y contraseñas compartidas para firmar el mensaje.

3.2.2.- Modelo de comunicación

Carriots dispone de *triggers* o *listeners* que son llamados nada más recibir un dato en la plataforma. Esto permite enviar la información a otras aplicaciones que se encuentren dentro o fuera de *Carriots*. Esta plataforma dispone de 100 emails gratuitos diarios, 5 SMS, 500 tramas, 1000 peticiones entrantes y 1000 peticiones salientes.

Como ejercicio extra, he implementado mi propia API con el objetivo de saltarme todas estas limitaciones. La última secuencia recibida desde *Carriots* (o desde la maqueta en modo de conexión directa), puede consultarse en http://www.smartblue.es/maqueta/api/public/secuencia.txt

3.2.3.- API de comunicación

La API externa de *Carriots* permite a las aplicaciones cliente comunicarse con la plataforma. Admite formatos en XML y JSON, y los verbos GET, POST, PUT y DELETE que permiten recibir información, crear un elemento, actualizarlo o eliminarlo, respectivamente.

Por otro lado, es necesario implementar cabeceras o *headers* que definen determinados comportamientos. En una solicitud, *Carriots* dispone de los siguientes *headers*:

- **Host:** Indica donde conectarse. Es de carácter obligatorio.

  Host: api.carriots.com
- **Carriots.Apikey:** Define la clave o *apikey*. Es de carácter obligatorio.

carriots.apiKey: ed5620c5013d782fe2eeaf9fd03b7fc0a42fafb06f0608b58d5ab2f9027f955f
- **Accept:** Define el tipo de solicitud MIME esperada. Es de carácter obligatorio. *Carriots* admite cuatro tipos distintos:

  Accept: application/json

  Accept: application/xml

  Accept: application/vnd.carriots.v2+json

  Accept: application/vnd.carriots.v2+xml
- **User-Agent:** Define el software del cliente. Es de carácter obligatorio.

  User-Agent: device A001

- **Content-Length:** Define la longitud del contenido en bytes. Es de carácter obligatorio.

  Content-Length: 209
- **Content-Type:** Define el tipo de contenido de la solicitud MIME. Es de carácter obligatorio. *Carriots* admite cuatro tipos distintos:

  Content-Type: application/json

  Content-Type: application/xml

  Content-Type: application/vnd.carriots.v2+json

  Content-Type: application/vnd.carriots.v2+xml

Mientras que en una respuesta, los *headers* disponibles son:

- **Date:** Devuelve la fecha y la hora del mensaje.
- **Server:** Contiene información sobre el software utilizado por el servidor. El valor predeterminado es: *Carriots* REST API.
- **Allow:** Lista de acciones permitidas para un recurso dado.
- **Connection:** Define las opciones de conexión para el emisor. *Carriots* siempre cierra la conexión después de obtener una respuesta.
- **Cache-Control:** Define las directivas para el manejo de la caché.
- **Content-Length:** Define la longitud de los datos en bytes.
- **Content-Type:** Define el tipo MIME para el cuerpo de respuesta.

3.3.- Interfaces I/O

3.3.1.- UART/USART

UART **[48]** responde por *Universal Asynchronous Receiver-Transmitter* o Transmisor-Receptor Asíncrono Universal, por sus siglas en inglés. Se encarga de controlar los puertos y dispositivos serie y de convertir los datos de paralelo a serie y viceversa para que puedan ser transmitidos.

El microchip ATmega2560 dispone de 4 de estos puertos. El valor típico de velocidad para comunicarse con el ordenador es de 9600 baudios, aunque se pueden introducir otros valores más elevados. De hecho, para una comunicación serie con el módulo ESP32 deberá utilizarse una velocidad de 115200 baudios.

	Serial	**Serial1**	**Serial2**	**Serial3**
Tx	1	18	16	14
Rx	0	19	17	15

Tabla 8: Pines de los puertos UART/USART en Arduino Mega 2560

UART no dispone de señal de reloj, por lo que deberá configurarse la misma velocidad de reloj tanto en el emisor como en el receptor para que los datos transmitidos sean correctamente interpretados en el destino. Sin embargo, también existe el protocolo USART **[49]**, el cual admite señales síncronas, además de múltiples protocolos tales como IrDA, LIN, Smart Card, DE (Driver Enable) para la interfaz RS-485, y Modbus, por mencionar algunos.

3.3.2.- SPI

El bus SPI **[50] [51]** (*Serial Peripheral Interface* o Interfaz de Periféricos Serie, por sus siglas en inglés), es un estándar de comunicaciones utilizado para controlar dispositivos electrónicos digitales que admitan un flujo de bits serie regulado por un reloj. Se trata de un dispositivo síncrono y está compuesto por cuatro señales:

- **SCLK (*Synchronous Clock*):** Es la señal de reloj. Con cada pulso se transmite un bit o *Takt*, en alemán.
- **MOSI (*Master Output Slave Input*):** Representa la salida de datos del *Master* y la entrada de datos al *Slave*.
- **MISO (*Master Input Slave Output*):** Representa la entrada de datos al *Master* y la salida de datos del *Slave*.
- **SS/Select (*Slave Select*):** Selecciona un esclavo.

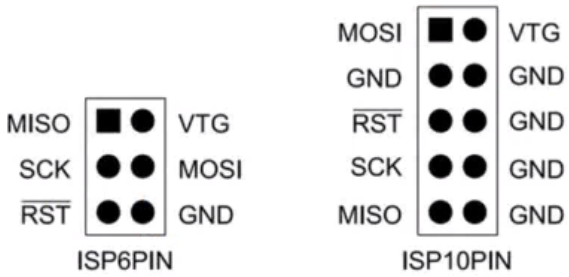


Figura 24: Conector ISP en versiones de 6 y 10 pines respectivamente

Los pines ICSP **[52]** (*In Circuit Serial Programming* o Programación Serie en Circuito, por sus siglas en inglés), permiten grabar el *bootloader* en el microcontrolador o modificar el programa a través de este puerto sin necesidad de sacarlo del zócalo. El pin 1 se puede identificar mediante un pequeño punto blanco que viene impreso.

	MOSI	**MISO**	**SCK**	**SS (Esclavo)**
Puerto	51	50	52	53
ICSP	4	1	3	-

Tabla 9: Pines de los puertos SPI en Arduino Mega 2560

	MOSI	MISO	SCK	SS (Esclavo)
Uno	11	12	13	10
Nano	11	12	13	10
Mini Pro	11	12	13	10

Tabla 10: Pines de los puertos SPI en otros Arduinos

Además de ser un puerto para programar Arduino, también es el conector de expansión del bus SPI utilizado para comunicarse con periféricos o *shields* de expansión.

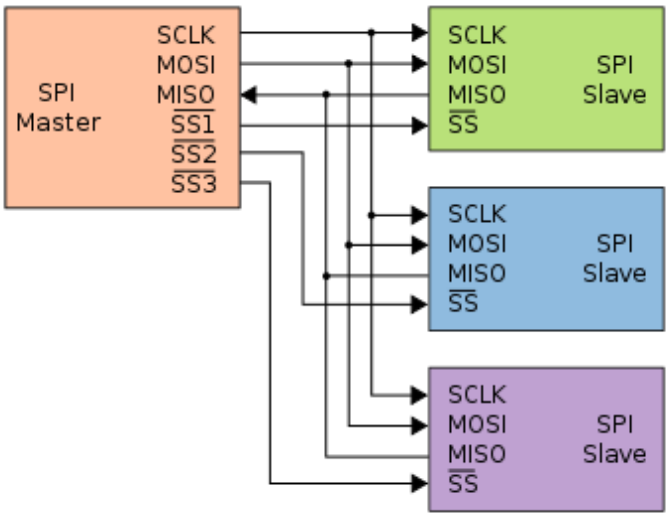


Figura 25: Conexión SPI con un master y tres esclavos

3.3.3.- I2C

El bus I2C **[53] [54]**, también conocido como IIC, I²C o TWI (*Two Wire Interface* o Interfaz de Dos Hilos, por sus siglas en inglés), es una norma de comunicación digital propuesta por Phillips a principios de los 80. Está formado por los pines SDA para transmitir los datos, y el pin SCL, un reloj asíncrono que indica cuando leerlos. Deben incluirse también unas resistencias de *pull-up* entre Vcc y los pines SDA y SCL.

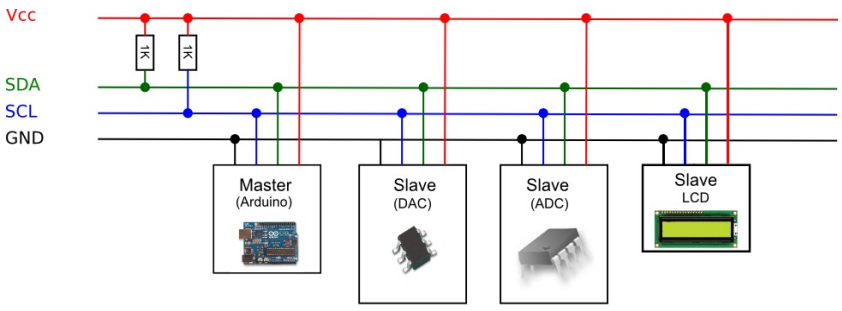


Figura 26: Diagrama de una conexión I2C

Cada dispositivo conectado a este bus tiene una dirección única de 7 bits, por lo que se pueden conectar hasta 128 dispositivos, siendo al menos uno de ellos el master, y el resto esclavos.

Arduinos	SDA	SCL
UNO	A4	A5
Due	20	21
Mega 2560	20	21

Tabla 11: Pines de los puertos I2C en Arduino

La librería I2C en Arduino se llama Wire.h.

3.4.- Sensorización

Actualmente existen varias propuestas que solucionan el problema que se plantea. Las diferentes soluciones proponen la identificación de cada una de las plazas de estacionamiento con un sensor que detecta la presencia de un vehículo. Lo único que difiere entre una u otra propuesta es la forma en la que estos sensores detectan el vehículo y la manera en la que se interconectan entre sí. Los sensores pueden ser intrusivos o no intrusivos. Los primeros se colocan en pequeñas cavidades hechas en el suelo y se conectan por debajo de la calzada, lo cual requiere levantar parte del pavimento para su instalación; mientras que los segundos se adhieren al suelo y se conectan de forma inalámbrica. Este proyecto se enfocará en estos últimos debido a su bajo coste y fácil instalación y mantenimiento, con relación a los intrusivos.

3.4.1.- Sensores intrusivos

Sensores infrarrojos activos: Este método detecta los vehículos emitiendo energía infrarroja y comprobando la cantidad de energía reflejada. Para una medición más precisa, el sensor transmite múltiples haces. Sin embargo, un inconveniente es su sensibilidad hacia las condiciones medioambientales, tales como niebla o nieve, que afectarían a su correcto funcionamiento.

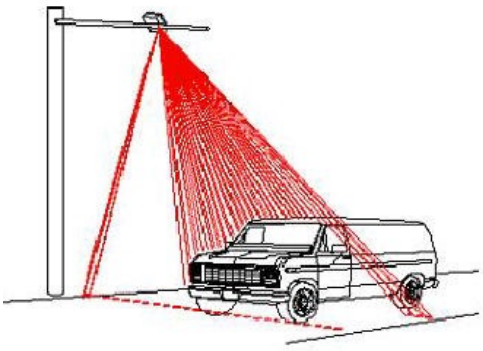


Figura 27: Detección de un vehículo mediante sensores infrarrojos activos

Detectores de bucle de inducción: Los detectores de bucle de inducción (ILD por sus siglas en inglés) son bucles de cable de diferentes tamaños los cuales son excitados con señales cuyas frecuencias varían entre 10 y 50 kHz. La frecuencia de oscilación de la bobina de inducción está controlada directamente por la inductancia del bucle que cambia con la presencia de un vehículo. Este método ha sido probado como uno de los más robustos y se trata de una tecnología ampliamente investigada y conocida en el mercado. Además, su flexibilidad permite el uso en una gran variedad de aplicaciones como semáforos o pasos a nivel. Se puede ampliar la zona de detección de vehículos combinando varios bucles juntos. En comparación con otras técnicas de uso común, este método ofrece la mejor precisión. De hecho, los sensores de bucle de inducción se han convertido en el estándar común para obtener mediciones precisas de ocupación.

Sin embargo, es un método caro de mantener. No sólo se requieren múltiples detectores para controlar una ubicación específica, sino que también son sometidos a un desgaste debido al tráfico y la temperatura. Su precisión de detección también se vería comprometida cuando el diseño requiriese de operar con una gran cantidad de vehículos. Es insensible a las condiciones climáticas, tales como la lluvia, niebla o nieve, pero no al agua acumulado, especialmente si el pavimento está agrietado.

Magnetómetro de saturación: Este dispositivo funciona mediante la detección de una perturbación (anomalía magnética) en el campo magnético horizontal y vertical de la Tierra. Los magnetómetros de saturación ofrecen la ventaja de ser insensibles a las condiciones climáticas, tales como la nieve, la lluvia o la niebla. También son más precisos y menos susceptibles a error que los bucles de inducción con una gran cantidad de tráfico. Entre las desventajas del uso de estos dispositivos se encuentra la necesidad de proximidad con el vehículo para una detección más precisa.

Figura 28: Paso a nivel con barrera regulado por un magnetómetro de saturación

Magnetómetro de inducción o *search coil*: Este método identifica la presencia de un vehículo midiendo el cambio en las líneas de flujo magnético causado por el vehículo en movimiento según la ley de Faraday de la inducción. Al igual que el magnetómetro de saturación,

es insensible a las condiciones climáticas, tales como la nieve, la lluvia y la niebla. Además, es menos susceptible a fallos por la gran cantidad de tráfico que los bucles de inducción. Para identificar los vehículos, se requieren diseños de sensores especiales y software de procesamiento de señal. Si bien es un sensor de intrusión, algunos modelos pueden ser instalados sin la necesidad de levantar el pavimento.

Sensores magnetorresistivos: Este grupo abarca los sensores de magnetorresistencia ansiotrópica (AMR), los sensores de magnetorresistencia gigante (GMR), los sensores de unión de túnel magnético, los sensores de magnetorresistencia extraordinaria y los sensores de magnetorresistencia balística, los cuales funcionan con una corriente constante. Estos sensores son ligeros y pequeños haciendo que sean versátiles y fáciles de implementar. Funcionan en un rango de temperatura que va desde los -55 hasta los 200 grados centígrados y son dispositivos de bajo coste. Se utilizan ampliamente para la detección de vehículos y son sensibles a la posición y orientación.

Figura 29: Sensores magnetorresistivos en la calzada

Sensores piezoeléctricos: Están hechos de un material capaz de convertir la energía cinética en energía eléctrica cuando se somete a una vibración o un impacto mecánico. Se puede diferenciar un vehículo con extrema precisión al recopilar información adicional. Además, proporciona una lectura más precisa de la velocidad del vehículo y los clasifica en función del peso y la distancia entre ejes. Una de sus desventajas es la necesidad de utilizar múltiples detectores para una única ubicación. Además, también es extremadamente sensible a la alta temperatura y el tráfico.

Sin embargo, este sistema puede ser de gran utilidad para aprovechar las vibraciones de los camiones y automóviles a su paso por la calzada recogiendo esta energía en unas baterías que posteriormente pueden utilizarse para alimentar otros dispositivos, tal y como hizo la empresa Innowatech en Israel.

Mangueras neumáticas: Detectan el vehículo mediante la presión de aire creada, produciendo señales cuando un vehículo pasa o se detiene sobre una manguera neumática. A pesar de que ofrecen una solución de bajo coste, así como una instalación rápida y de fácil mantenimiento, estos sensores son sensibles a la temperatura y deben estar bien fijados para evitar que los vehículos los arrastren o los rompan. Su uso sólo es viable para uso temporal y en tránsitos fluidos.

Figura 30: Mangueras neumáticas

Peso en Movimiento (WIM): Estos sensores son capaces de detectar el peso del vehículo. Aunque inicialmente fueron pensados para pesar el vehículo sin detenerse (*Weight In Movement*), también pueden detectar la presencia de un vehículo por su propio peso.

3.4.2 Sensores no intrusivos

Radar de microondas: Transmiten energía (1-30 GHz) a través de una antena y detectan los vehículos por la energía reflejada de nuevo hacia la antena. Existen dos tipos de sensores de radar de microondas los cuales son: Radar de Onda Continua (CW) y Radar de Onda Continua de Frecuencia Modulada (FMCW). Estos sensores ofrecen la ventaja de ser insensibles al clima. También son capaces de recoger algunos datos tales como el flujo del tráfico o la velocidad de cada vehículo. Su principal desventaja reside en la necesidad de un segundo sensor Doppler imprescindible para detectar un vehículo detenido.

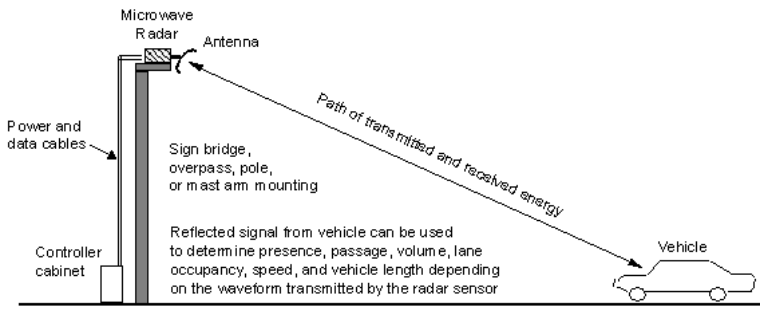


Figura 31: Funcionamiento de un radar de microondas

Sensores acústicos: Los sensores acústicos detectan un vehículo mediante la energía acústica emitida por el mismo y recogida por micrófonos instalados para este fin. Las ventajas ofrecidas por estos sensores son la capacidad de detectar múltiples vehículos así como la detección pasiva. Además son insensibles a la lluvia, no tanto así a las bajas temperaturas.

Sensores infrarrojos pasivos: Estos sensores identifican el estado de ocupación de una plaza de aparcamiento mediante la detección de cambios en la energía emitida por el vehículo y el pavimento. También pueden medir la velocidad del vehículo. Sin embargo, la sensibilidad de estos sensores se ve reducida bajo la lluvia intensa, la nieve y la niebla densa.

RFID: Se trata de una tecnología segura y eficiente que consta de transceptor, transpondedor y antena. El transceptor se utiliza para enviar y leer información de la unidad de transpondedor, que contiene la información codificada, a través de la antena. Los transpondedores activos pueden reprogramarse mediante conexión inalámbrica, mientras que los pasivos tienen una vida útil ilimitada. El uso de estos sensores ofrece una instalación de bajo coste y mantenimiento. La dificultad reside en la necesidad de colocar transpondedores en todos los vehículos, así como otros temas vinculados con la privacidad de los usuarios.

Sensores ultrasónicos: Estos sensores transmiten pulsos de entre 25 a 50 kHz al aire y trata de detectar la energía reflejada hacia el sensor. Junto con un módulo de procesamiento de señales, las energías ultrasónicas reflejadas se analizan para detectar la ocupación de una plaza de estacionamiento. Tienen la ventaja de detectar vehículos que exceden cierto límite de altura y son fáciles de instalar. Sin embargo, los cambios de temperatura y las altas turbulencias afectan al rendimiento del sensor, aunque algunos modelos cuentan con una compensación de temperatura incorporada.

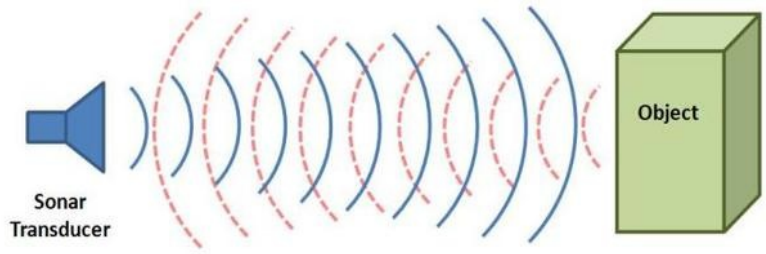


Figura 32: Funcionamiento de un sensor ultrasónico

Procesador de vídeo: Un procesador de vídeo consiste en una o varias cámaras, software para la interpretación de imágenes y un ordenador basado en microprocesador para la digitalización y procesamiento de las imágenes recogidas. El análisis cuidadoso de cuadros continuos capturados por el procesador de imagen de vídeo puede detectar vehículos, ya que revela diferencias entre

tramas subsiguientes. Las zonas de detección pueden modificarse fácilmente. Además se puede adaptar fácilmente a cualquier forma geométrica para optimizar su funcionamiento.

Este sistema sólo es rentable si se requieren de muchas zonas de detección en el campo de visión de la cámara y se dispone de una buena iluminación. El tiempo meteorológico, las sombras, la proyección de vehículos en carriles adyacentes, el contraste del día a la noche, el agua, la suciedad, la sal, los carámbanos y las telarañas en la lente de la cámara pueden afectar al rendimiento considerablemente. Además, los movimientos de la cámara debido a fuertes vientos también podrían agravar el problema.

Capítulo 4

Propuesta del proyecto

4.1.- Requisitos del sistema

Con este proyecto se pretende que el usuario sea capaz de encontrar un sitio donde estacionar su vehículo lo antes posible, lo cual genera valores como la rapidez, el ahorro o la sostenibilidad.

- Rapidez: Al estar conectado mediante una *app* en tu móvil, permite ver desde antes de salir de tu casa todas las zonas libres donde aparcar. Gracias a este sistema, puedes distribuir tu tiempo. Ajustar con mayor precisión la llegada a tu cita o trabajo. Esto ayudará a estar más tiempo con tu familia y contribuir a la conciliación laboral.

- Ahorro: El sistema de parking inteligente generará un ahorro en combustible ya que al guiarte a la zona de parking libre, se evitarán rodeos innecesarios en la búsqueda, pero también se implementarán otros ahorros como en desgaste de neumáticos o en tiempo.

- Sostenibilidad: Actualmente existe una gran preocupación por la sostenibilidad del medio ambiente. Sobretodo en las grandes ciudades como en Madrid o Barcelona que para evitar los episodios de contaminación restringen la circulación de vehículos y una de estas medidas es la prohibición de aparcar en las zonas de alta contaminación. Esto es así porque un vehículo buscando parking se desplaza innecesariamente incluso varios kilómetros. Este sistema busca que esta acción sea más rápida, y en consecuencia se emitirían menos gases de efecto invernadero a la atmósfera, reduciendo el número de episodios de alta contaminación. Asimismo, se consigue una reducción del ruido.

Aplicando estos tres valores vamos a resolver los siguientes problemas:

- A los usuarios: Encontrar una plaza libre donde aparcar, ya sea en la calle, (zona azul o no), como en el parking de un centro comercial, grandes centros de compra, plazas de garaje de particulares o de hoteles que pueden ponerlas en alquiler para darles rendimiento económico.

- A la administración: Que circulen menos coches y durante menos tiempo descongestionando las ciudades, ahorrando combustible y reduciendo las emisiones contaminantes que crean los episodios de alta contaminación de grandes ciudades teniendo que restringir la circulación con las pérdidas que esto conlleva, tanto en el sector público como en el privado, además de una gestión más eficiente de las zonas azules en las cuales se podrá obtener más beneficio y menos incidencias, que son, mayoritariamente, las que hacen fracasar el sistema de estacionamiento en áreas restringidas o zonas azules.

- Al sector privado: Que sus trabajadores no lleguen tarde y cumplan mejor con su jornada laboral. Que puedan alquilar sus plazas de parking obteniendo una mayor rentabilidad, como por

ejemplo, en hoteles con alta capacidad de estacionamiento pero la mayoría de sus plazas suelen encontrarse libres. También se puede implementar este sistema en grandes superficies con mucha disponibilidad de parking. Al dirigir al usuario a una plaza libre siempre lo más cerca posible de la entrada al local comercial, se reduce el tiempo de búsqueda de parking y se incrementa el tiempo en la gran superficie, aumentando la probabilidad de consumo.

Con todo lo anterior, se resuelve la necesidad que conlleva la búsqueda de plaza de estacionamiento para cualquier usuario reduciendo tiempo y contaminantes. Para la administración se resuelve el problema de una gestión eficiente de las zonas azules y se reducen los episodios de alta contaminación. Y para las empresas privadas, se pone en valor una plaza de parking que de otro modo no generaría ningún beneficio.

Este sistema novedoso de parking inteligente desempeña una finalidad tanto económica como para el medio ambiente, ya que ayuda al usuario a encontrar estacionamiento más rápidamente y a la administración a reducir la contaminación. El diseño puede ser muy personalizado y adaptable a cada usuario. Por otra parte, el precio por parte de los usuarios puede verse disminuido ya que aumenta la eficiencia de la gestión en zonas azules. Al mismo tiempo, a la administración les reduce costos en mano de obra y vigilancia. A su vez, la implantación de este sistema es capaz de reducir riesgos como colisiones, atropellos o alcances asociados a las distracciones por la búsqueda de un estacionamiento.

4.2.- Modelos de negocio asociados al sistema

Este sistema tiene diferentes formas de ingreso por sus distintas líneas de negocio y clientes.

- **Cliente particular:** Nuestro cliente particular estaría dispuesto a pagar un coste por el servicio que le fuera lo suficientemente atractivo, sobretodo para poder generar unos beneficios a medio plazo, ya que la finalidad de la instalación consiste en poner en valor una plaza de estacionamiento que no utiliza o sólo lo hace durante un determinado periodo de tiempo al año, mes o incluso del mismo día.

En la actualidad, la forma en la que se pone en valor este tipo de estacionamientos es con un alquiler mensual o anual cerrado a un precio descrito en el contrato y pactado de antemano. Esto implica que a lo largo del periodo contratado sea difícil cambiar las condiciones firmadas. Este sistema pretende que este alquiler sea más fluido y fácil, ya que sólo hace falta el registro una única vez, dando lugar a que el estacionamiento no esté sujeto tan rígidamente a un contrato de este tipo. El método de pago más común para este tipo de servicio es el pago de la mensualidad a través de transferencia bancaria. Esto obliga, en algunos casos, a desplazarse al banco para realizar el ingreso. En otros, el pago se efectúa en mano, favoreciendo así la economía sumergida.

Con el nuevo sistema, se puede pagar a través de la *app* o de la web cumplimentando un formulario con los datos bancarios, que será el mismo en el cual se recibirán los ingresos generados por la plaza de estacionamiento. Para poder acceder al recinto, se mostrará un código Bidi en la aplicación móvil que accionará el mecanismo de apertura a través de un lector de códigos Bidi instalado a la entrada del parking.

- **Ayuntamiento:** En esta ocasión podemos tener dos tipos diferentes de modelo de negocio. El primero es similar al de un cliente particular: se hace una instalación y se cobra una cuota mensual, y la gestión recae en el propio ayuntamiento.

En la actualidad, la zona azul se paga utilizando unas máquinas situadas al principio de cada calle, en las cuales el cliente tiene que calcular cuánto tiempo va a estar estacionado su vehículo y abonar el importe correspondiente. Esto provoca que en ocasiones se pague de más por el servicio, pero también que se pueda pasar de la hora dando lugar a multas. La solución adoptada consiste en pagar por el tiempo que se está estacionado, ya que todo el proceso será controlado de manera automática a través de la aplicación y los sensores en la plaza de estacionamiento. Por lo tanto, el funcionamiento sería muy parecido al de un parking público de pago.

Muchos ayuntamientos utilizan la zona azul en sus calles de forma disuasoria para los estacionamientos prolongados, es decir, pretenden una continua movilidad de vehículos para que siempre exista algún sitio libre. El sistema da una solución a este problema con una tarifa de precios dinámicos que se incremente a forma de penalización cuanto más tiempo se ocupe la plaza. Este modelo de negocio posee una diversa gama de tarifas a negociar con el ayuntamiento.

El segundo modelo de negocio es la subcontrata, al igual que se hace con la limpieza de las calles, el sistema de alquiler de bicicletas o el de coches eléctricos. Por lo tanto, el usuario final es estrictamente nuestro cliente.

En cualquiera de los dos casos, el terminal de pago debería mantenerse en la calle, al menos durante un tiempo, para aquellos usuarios que se resistan o tengan barreras tecnológicas. Para la implantación de este sistema en un ayuntamiento, al ser un servicio más eficiente, el precio por minuto se puede mantener, aumentando la recaudación.

- **Hoteles:** Otro segmento de clientes hace referencia a los hoteles como pudiera ser la cadena hotelera NH. Este sistema es similar al del cliente particular, pero a diferencia de este, el hotel posee multitud de plazas de estacionamiento. La cadena estaría dispuesta a contratar el servicio a un coste que le fuera lo suficientemente interesante para obtener beneficios a medio plazo. Por lo tanto, la tarifa mensual aplicable no distará mucho de la del cliente particular, aunque al tratarse de más dispositivos, su tarifa irá en función del volumen.

Hoy en día cualquier hotel sólo ofrece sus estacionamientos a los clientes alojados, pero la mayoría no hace uso de este servicio. De ahí se abre una necesidad de la cual este sistema puede ofrecer una solución viable. La metodología de pago será por cuantía fija mensual en función del número de plazas que se deseen cubrir.

- **Centros comerciales y grandes superficies:** Estas estarían dispuestas a pagar un precio directamente relacionado con el aumento de ventas gracias a la racionalización que este sistema crea del uso de cada estacionamiento. En la actualidad la zona de parking no supone ningún tipo de gasto, salvo aquellos derivados del mantenimiento.

Como se ha observado en las anteriores descripciones de clientes, el ingreso general viene dado mayoritariamente de la cuota mensual de este servicio, ya fuera un cliente particular, distinto tipo de contrato con el ayuntamiento, cadena hotelera o centro comercial o gran superficie. Aunque la cuota mensual sea la fuente principal de ingresos, otras vienen derivadas de la instalación del paquete, mantenimiento, alquiler de los dispositivos y el uso directo del servicio.

4.3.- Entorno Software

4.3.1.- Plataforma *Carriots*

Carriots es una Plataforma como Servicio (PaaS, por sus siglas en inglés), diseñada para proyectos del Internet de las Cosas (IoT) y de Máquina a Máquina (M2M). **[55]**

Una plataforma IoT es el software de soporte que conecta hardware, puntos de acceso y redes de datos a las aplicaciones de usuario final mediante técnicas SaaS, IaaS y PaaS. **[56]** Existen infinidad de proveedores de IoT **[57]** entre los que se encuentran Amazon **[58]**, IBM **[59]** o Google **[60]**. Actualmente este mercado se encuentra en auge y se prevé que en 2019 mueva cerca de un billón de dólares y 1,6 billones en 2021. Estas plataformas son necesarias para soluciones de middleware o lógica de intercambio de información entre aplicaciones. Deben estar basadas en la nube y tener la capacidad de ser gestionadas, mantenidas y accesibles desde cualquier lugar. **[61]**

Una plataforma de IoT consta de 8 bloques: **[62]**

1.- **Conectividad y normalización:** Garantiza la transmisión de los datos y la interacción con todos los dispositivos.

2.- **Gestión de dispositivos:** Garantiza un correcto funcionamiento del sistema.

3.- **Base de datos:** Garantiza el almacenamiento escalable de un volumen de datos significativo.

4.- **Procesamiento y gestión de la acción:** Garantiza el control de reglas de acción de evento-disipadores que permiten la ejecución de determinadas acciones basadas en los datos recogidos por los sensores.

5.- **Analítica:** Garantiza un análisis complejo de la agrupación de datos básicos y un aprendizaje continuo y automático.

6.- **Visualización:** Garantiza el acceso a los datos representados a través de gráficos.

7.- **Herramientas adicionales:** Facilita a los desarrolladores probar y comercializar sus servicios además de visualizar, gestionar y controlar todos los dispositivos conectados.

8.- **Interfaces externas:** Facilita una interconectividad con sistemas de terceros a través de interfaces de programación de aplicaciones (API), kits de desarrollo de software (SDK) y puertas de enlace.

Para el desarrollo de este proyecto se va a utilizar la plataforma *Carriots* ya que cubre ampliamente los requisitos necesarios. Se pueden enviar tramas a *Carriots* con solicitudes HTTP POST. Para ello se debe crear esta solicitud con los datos de configuración específicos para esta plataforma, además del dispositivo previamente registrado y una APIKEY con los permisos de escritura.

```
POST /streams HTTP/1.1
Host: api.carriots.com
Accept: application/json
User-Agent: Arduino-Carriots
Content-Type: application/json
carriots.apikey: YOUR APIKEY HERE
Content-Length: YOUR CONTENT LENGTH HERE
Connection: close
```

Figura 33: Esquema de una solicitud HTTP mediante el método POST

A continuación se deben especificar los datos que se desean enviar.

```
{
"protocol":"v2",
"device":"YOUR DEVICEs ID_DEVELOPER HERE",
"at":1356390000,
"data":{"light":"YOUR DATA HERE"}
}
```

Figura 34: Esquema del envío de datos

Previamente habrá que dar de alta un nuevo dispositivo de los dos que permite la versión gratuita de la plataforma. Para ello se rellenan los datos de la siguiente figura:

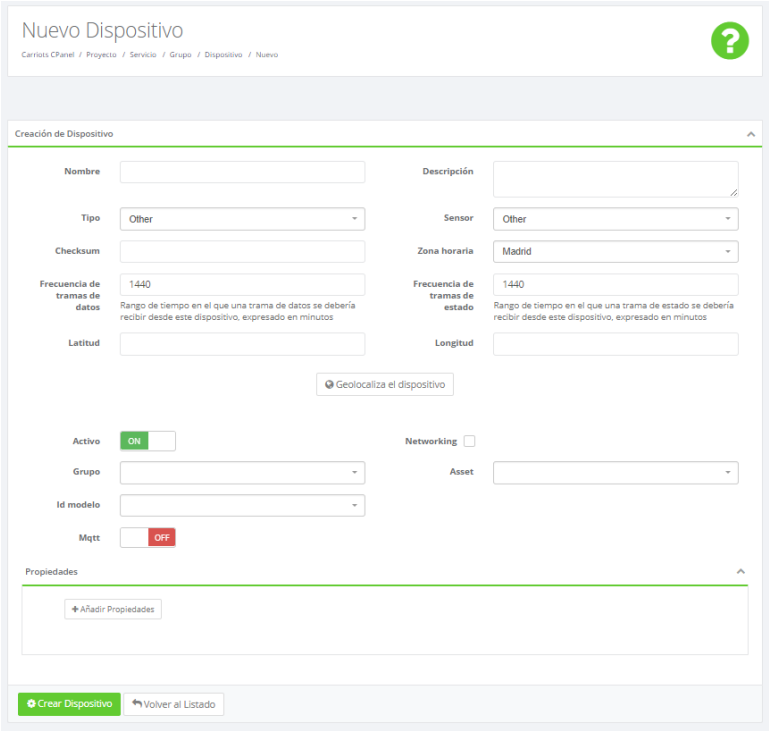


Figura 35: Registro de un nuevo dispositivo en *Carriots*

En el apartado de configuración se pueden encontrar las *Apikeys* disponibles que no son más que códigos de autorización con diferentes privilegios.

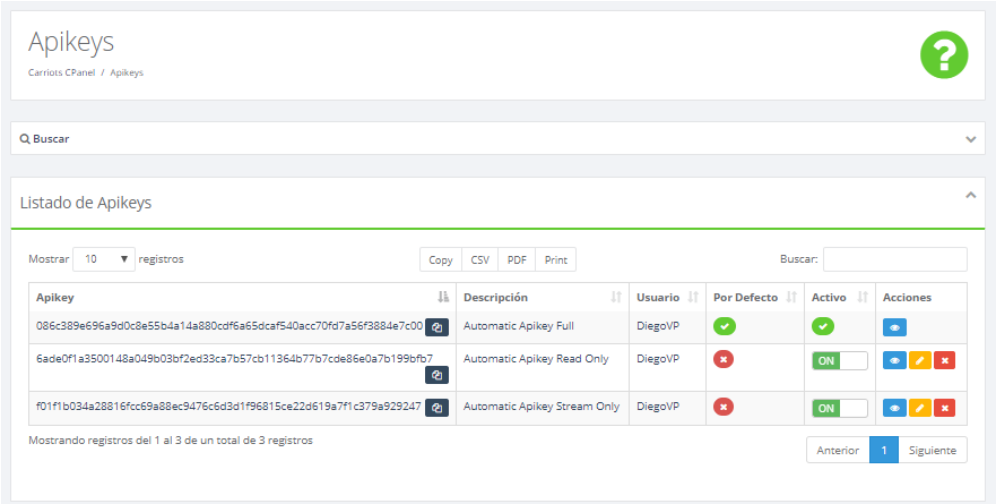


Figura 36: Listado de *Apikeys*

Finalmente las tramas enviadas se pueden visualizar en el apartado Datos.

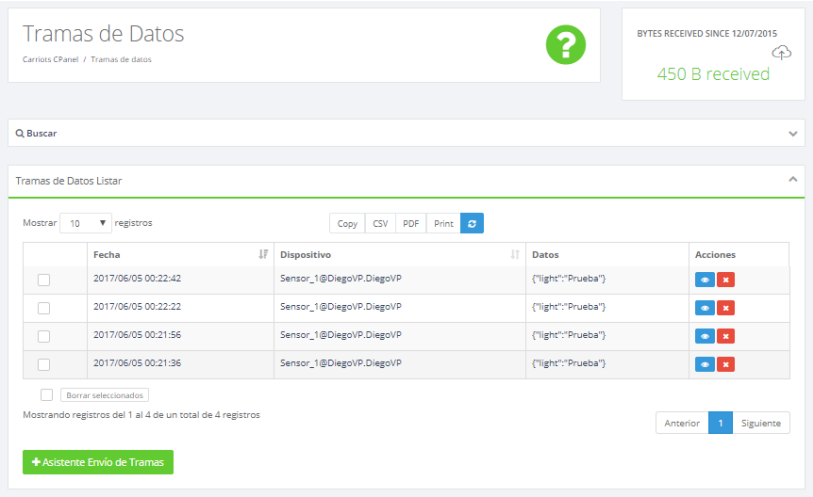


Figura 37: Listado de tramas de datos

En función de los datos recibidos se pueden crear eventos que realicen determinadas acciones, como por ejemplo enviar un SMS, un email o publicar un tuit.

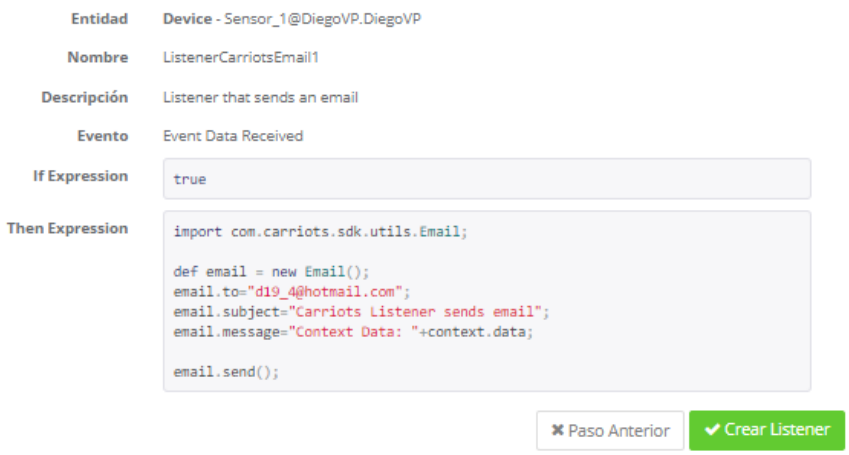


Figura 38: Crear un listener

4.3.2.- Software de Arduino (IDE)

El Entorno de Desarrollo Integrado o Entorno de Desarrollo Interactivo de Arduino (IDE, *Integrated Development Environment*, por sus siglas en inglés), es una aplicación informática que facilita al programador el desarrollo de software. **[63]** Contiene un editor de texto para escribir el código, una zona para mensajes, una consola de texto, una barra de herramientas con diferentes opciones y una serie de menús. Este software se conecta a la placa Arduino para subir el programa y comunicarse con él. **[64]**

Los programas escritos mediante el Software de Arduino se llaman *sketches*. Los *sketches* se escriben en el editor de texto y se guardan con extensión .ino (antes de la versión 1.0 se utilizaba la extensión .pde). En la consola inferior se pueden leer avisos o errores del programa, mientras que

la barra de herramientas permite verificar y subir los programas, además de crear, abrir y guardar *sketches*, y abrir el monitor serie. Este software está disponible para Windows, Mac OS X y Linux, así como también se puede encontrar online en la página web https://auth.arduino.cc para usuarios registrados.

Figura 39: Arduino IDE

Cada *sketch* está formado por dos funciones de tipo void. La primera de ellas denominada *setup()* es la encargada de configurar el proyecto y sólo va a ejecutarse una única vez, mientras que la segunda, *loop()*, se trata de un bucle infinito.

Para añadir funcionalidades extra al proyecto, se pueden incluir librerías al principio del *sketch*. Esto se hace mediante la etiqueta *#include*. También se pueden definir constantes con la etiqueta *#define*, o variables globales fuera de las funciones *loop()* y *setup()*.

Antes de subir cualquier *sketch* a una placa Arduino, es imprescindible seleccionar el puerto al que se encuentra conectada al ordenador, así como indicar el tipo de placa y su procesador. El lenguaje de programación de Arduino está basado en C++.

4.3.3.- Aplicación móvil

La aplicación móvil desarrollada es un espejo de la plataforma web sin opciones de configuración, es decir, simplemente muestra un mapa con el estado de las plazas en tiempo real. No tiene implementada ninguna otra funcionalidad.

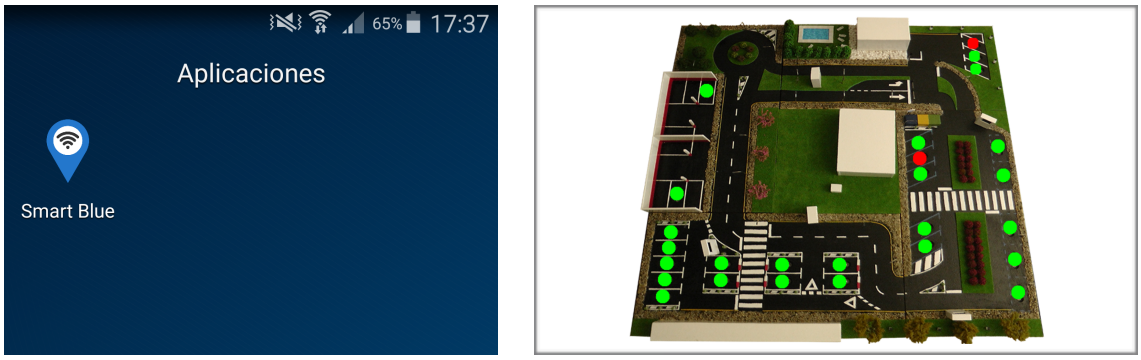


Figura 40: Capturas de la aplicación móvil

4.4.- Entorno Hardware

La configuración hardware que se persigue es la siguiente:

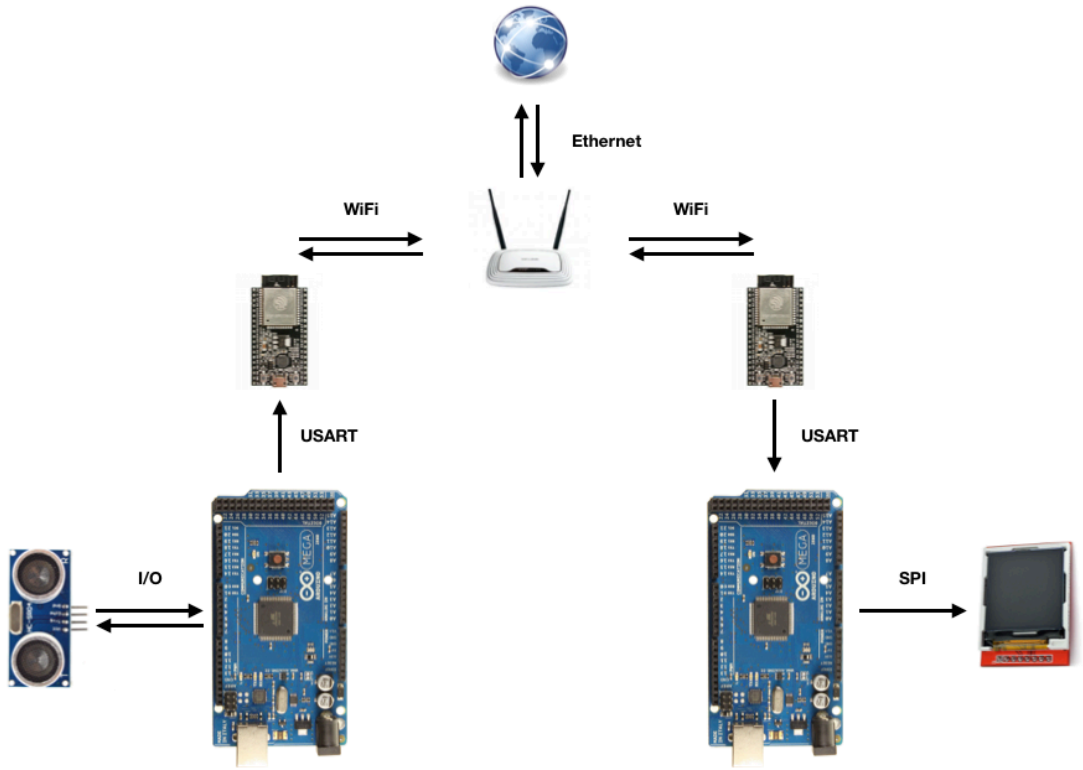


Figura 41: Esquema hardware de todo el sistema

Los 26 sensores HC-SR04 distribuidos por la maqueta emitirán un pulso ultrasónico a 40 KHz para detectar la presencia de un vehículo. Este dato será enviado al nodo central formado por un Arduino MEGA 2560 el cual comparará la secuencia obtenida con la almacenada previamente y si es distinta, la enviará al dispositivo ESP32 que se encuentra conectado con la plataforma *Carriots* mediante tecnología WiFi hasta su puerta de enlace predeterminada y tecnología Ethernet durante el resto del recorrido.

Una vez se reciban nuevas tramas de datos en la plataforma *Carriots*, esta emitirá un *trigger* a una página web externa (www.smartblue.es), la cual almacenará este valor en su base de datos y representará mediante una imagen canvas en HTML5 el estado de los sensores con círculos verdes o rojos sobreimpresionados en una fotografía de la maqueta.

Por otro lado, otro dispositivo ESP32 realizará peticiones a esta web externa cada 3 segundos para obtener la última trama en la que se muestra el estado de los sensores. Si esta trama difiere de la almacenada previamente, la enviará al otro Arduino MEGA 2560, el cual se encargará de enviar a la pantalla correspondiente el número de plazas actualizado haciendo uso del protocolo SPI.

Finalmente, este mapa con el estado de los sensores también se podrá consultar en una sencilla aplicación para Android.

4.4.1.- Arduino Mega 2560

Arduino es una plataforma italiana de hardware libre, basada en una placa compuesta por circuitos impresos que integran un microcontrolador con puertos analógicos y digitales de entrada y salida que pueden ser conectados a placas de expansión o *shields*, las cuales amplían las características de su funcionamiento. Asimismo, Arduino también es un entorno de desarrollo diseñado para facilitar el uso de la electrónica en proyectos multidisciplinares. **[65]**

Su nombre viene dado por el Rey Arduino, rey de Italia entre los años 1002 y 1014, y nombre del bar, el Bar di Re Arduino, donde uno de sus fundadores, Massimo Banzi, pasaba algunas horas. **[66]**

La primera placa Arduino fue introducida en 2005 con microcontroladores AVR de 8 bits integrados, pero no fue hasta octubre de 2012 cuando se incorporaron nuevos modelos de microcontroladores basados en el diseño Cortex M3 con arquitectura ARM de 32 bits.

Los microcontroladores AVR se basan en una arquitectura Harvard, mientras que ARM se basaba en von Neumann hasta la versión 9 que adoptó la arquitectura Harvard. **[67]**

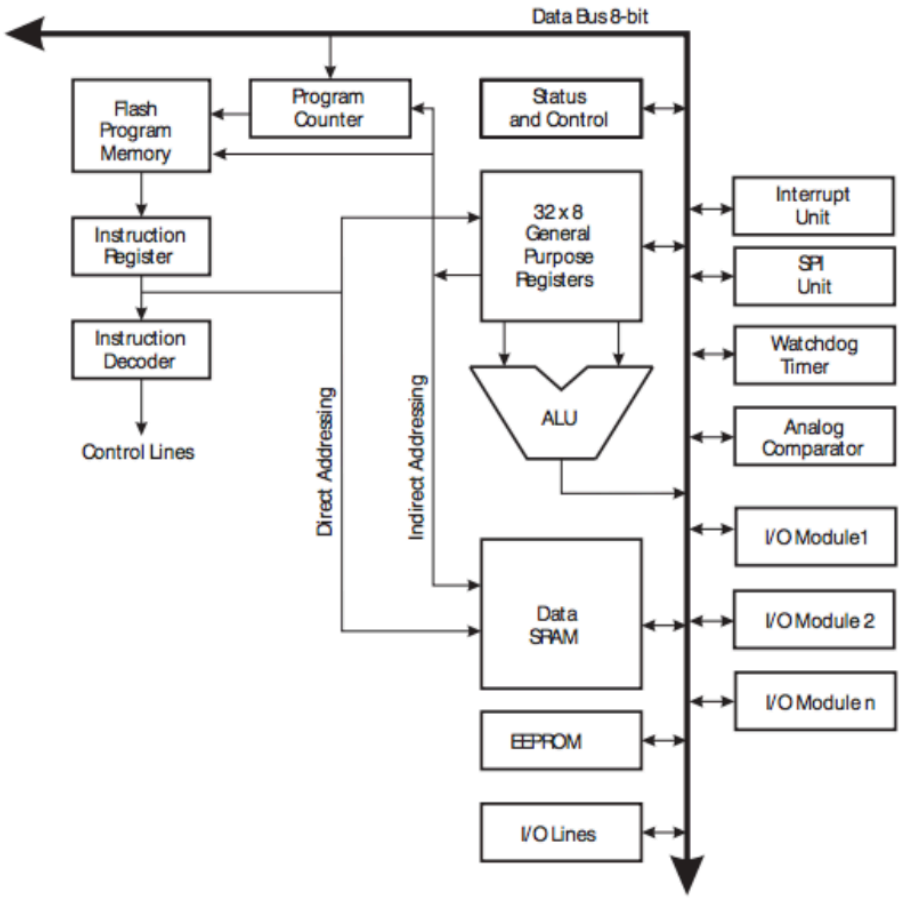


Figura 42: Diagrama de bloques de una arquitectura AVR

Característica de Arduino	UNO	Leonardo	Mega 2560	DUE
Tipo de microcontrolador	ATmega 328P	ATmega 32U4	ATmega 2560	ATSAM3X8E
Velocidad de reloj	16 MHz	16 MHz	16 MHz	84 MHz
Pines digitales E/S	14	20	54	54
Entradas analógicas	6	12	16	12
Salidas analógicas	0	0	0	2 (DAC)
Memoria de programa (Flash)	32 kb	32 kb	256 kb	512 kb
Memoria de datos (SRAM)	2 kb	2,5 kb	8 kb	96 kb
Memoria auxiliar (EEPROM)	1 kb	1 kb	4 kb	0 kb

Tabla 12: Comparativa de diferentes Arduinos

Existen más de 20 placas de Arduino diferentes, entre las que se encuentran la archiconocida Arduino UNO, Arduino Leonardo o Arduino Mega 2560, todas ellas con microcontroladores Atmel de 8 bits. Además, también existen cerca de una decena de placas de expansión o shields capaces de ofrecer conectividad GSM, Ethernet o WiFi, entre otras funcionalidades, a cada una de estas placas.

La placa Arduino Mega 2560, basada en el microcontrolador ATmega 2560, ha sido diseñada para proyectos complejos. Dispone de 54 pines digitales de entrada o salida (de los cuales 15 pueden ser utilizados como salidas PWM, *Pulse-Width Modulation* o Modulación por Ancho de Pulso, por sus siglas en inglés), 16 pines analógicos de entrada, 4 UARTs (*Universal Asynchronous Receiver-Transmitter* o Transmisor-Receptor Asíncrono Universal, por sus siglas en inglés, que controla los puertos y dispositivos serie **[68]**), un reloj de 16 MHz y un mayor espacio para el almacenamiento del programa. Es además compatible con la mayoría de placas de extensión diseñadas para Arduino UNO y las anteriores placas Duemilanove o Diecimila. **[69]**

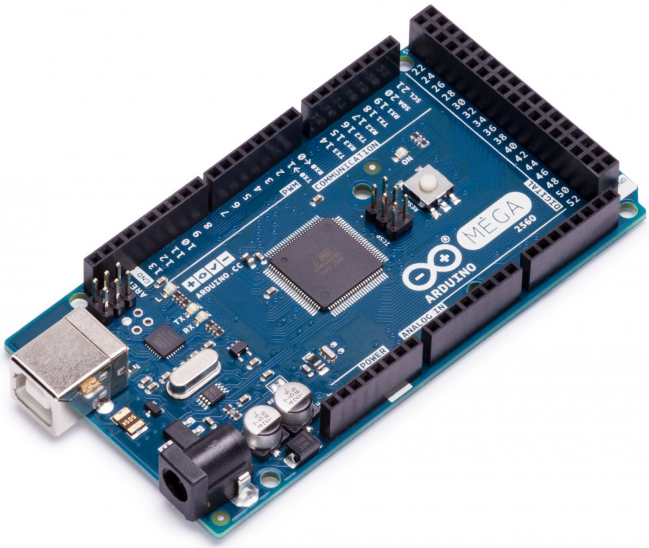


Figura 43: Arduino Mega 2560

4.4.2.- Módulo ESP32

ESP32 **[70]** es el nombre mediante el cual *Espressif* comercializa una familia de módulos inalámbricos sucesores del ESP8266, capaces de realizar comunicaciones WiFi en la banda de 2,4 GHz y Bluetooth. A diferencia de su antecesor, este transceptor dispone de un sensor de efecto Hall y otro de temperatura incorporados. Trabaja a 160 MHz y su microcontrolador es de doble núcleo.

El primero de estos chips se publicó en agosto de 2014 con el módulo ESP-01, desarrollado por la empresa Al-Thinker. Este dispositivo permite a otros microcontroladores conectarse a una red WiFi y realizar conexiones simples con el protocolo TCP/IP usando comandos AT.

Inicialmente fue diseñado como adaptador serie a WiFi, pero el nuevo dispositivo ESP32, que no apareció en el mercado hasta septiembre de 2016, es mucho más potente, no solo porque es más rápido, sino porque está diseñado pensando en que sea un microcontrolador para el IoT. **[71]**

En la tabla siguiente se muestra una comparativa entre ambos módulos inalámbricos:

Especificaciones	ESP8266	ESP32
Procesador	Xtensa Single-Core 32-bit L106	Xtensa Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n WiFi	HT20	HT40
Bluetooth	No	Bluetooth 4.2 e inferior
Frecuencia típica	80 MHz	160 MHz
SRAM	160 kb	512 kb
Flash	Flash por SPI, hasta 16 MB	Flash por SPI, hasta 16 MB
GPIO	17	36
Hardware / Software PWM	No / 8 canales	1 / 16 canales
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	No	1
Interfaz MAC Ethernet	No	1
Sensor de efecto Hall	No	Sí
Sensor de temperatura	No	Sí
Temperatura de trabajo	-40ºC – 125ºC	-40ºC – 125ºC

Tabla 13: Comparativa de ESP8266 vs ESP32

Con un procesador Xtensa Dual-Core LX6 de 32 bits a 160 ó 240 MHz, el ESP32 utiliza dos núcleos permitiéndole dedicar uno de ellos a la comunicación IP y WiFi y el otro al resto de procesos, resolviendo así uno de los mayores impedimentos con los que contaba la arquitectura de su antecesor.

Otro de los inconvenientes es la seguridad, lo cual limita su uso en aplicaciones industriales y comerciales. El ESP32 resuelve este problema ya que incorpora un acelerador de encriptación por hardware. Además, permite la encriptación del código en la memoria flash para que no sea posible leerlo ni realizar ingeniería inversa.

Figura 44: Módulo ESP32

Por último, dispone también de un coprocesador de muy bajo consumo asociado al RTC permitiendo realizar ciertas tareas mientras el chip se encuentra en modo de sueño profundo. Todo esto con un precio actual que apenas duplica el de su hermano pequeño.

4.4.3.- Paneles solares fotovoltaicos

Una célula solar es un dispositivo electrónico que convierte la luz en corriente eléctrica. Esta conversión es directa, sólo requiere de luz solar y no produce ningún residuo. Se busca que este sea un proyecto sostenible, por lo que en el proceso de validación se utilizarán dos placas solares que alimentarán el circuito. Cada una de ellas está compuesta por 12 celdas solares de silicio monocristalino **[72]**.

Según la hoja de características **[73]**, estas celdas tienen unas dimensiones de 125 mm × 125 mm ± 0,5 mm con un grosor de 200 µm ± 30 µm. Alcanzan una eficiencia de hasta el 17,49%, una diferencia de potencial máxima de 0,52 V y una corriente máxima de 4,98 A, lo que se traduce en una potencia máxima de 2,59 W cada una. En su conjunto, cada panel solar puede ofrecer una tensión máxima teórica de 6,24 V y 31,08 W de potencia. Al conectar ambos paneles en serie se pueden conseguir hasta 12,48 V y 62,15 W. Los mismos paneles en paralelo ofrecen 6,24 V y 9,96 A como máximo.

Las celdas de silicio se conectan entre sí con una cinta conductora o *tab wire* de 1,8 mm de ancho y 0,16 mm de grosor, intercalándolas a modo de acordeón (la parte superior de una celda se suelda a la cinta, y esta a su vez se suelda a la parte inferior de la siguiente celda). Se utilizan dos de estas cintas para cada celda realizando una conexión en serie. Para conformar el panel de 12 celdas, se sueldan dos tiras de 6 celdas cada una. Para ello se utiliza una cinta conductora más gruesa denominada *bus wire*, de 5 mm de ancho y 0,2 mm de grosor.

Por último, se debe encapsular cada panel con múltiples capas como se muestra en la siguiente imagen:

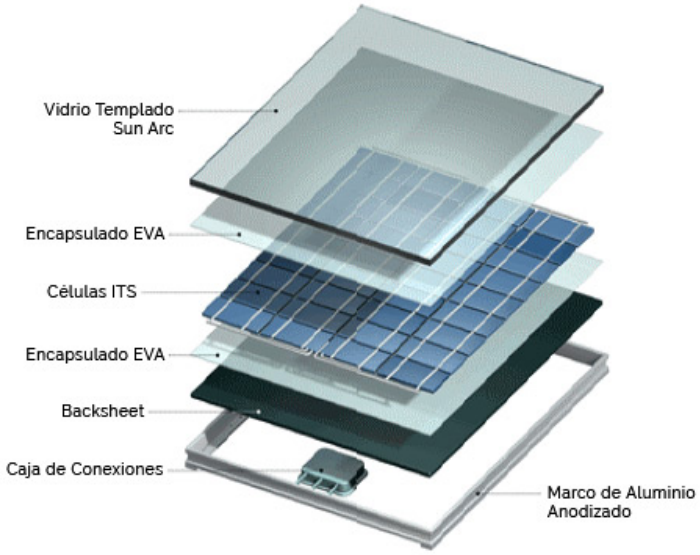


Figura 45: Partes de un panel solar

a) Diodo Schottky

Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido. **[74]**

Mientras exista radiación solar, las células producirán corriente que circulará en el sentido deseado, cargando la batería. Sin embargo, por la noche, al no haber radiación solar, la corriente circulará en sentido inverso, por lo que se descargará la batería. Para evitar este inconveniente, similar al proceso de fotosíntesis de las plantas, se debe colocar uno de estos diodos para bloquear la circulación de la corriente en sentido inverso. Estos diodos se denominan diodos de bloqueo que son de tipo Schottky ya que estos poseen una diferencia de potencial de apenas 0,3 voltios, a diferencia de un diodo común que tiene 0,7 V.

Por otro lado, puede ocurrir que una de las placas solares se estropee, se ensucie o sea cubierta por alguna sombra. Cualquiera de estas situaciones produciría un circuito abierto desconectando toda la serie de celdas. Para solucionar este problema, se debe incluir uno de estos diodos en cada panel solar, denominados diodos de *bypass*, cerrando así el circuito y permitiendo que, aunque un panel no funcione, permita que funcionen el resto de paneles.

Para el caso con mayor amperaje, al conectar ambas celdas en serie, se pueden obtener hasta 10 amperios. Y dado que la tensión máxima es de 12,48 voltios, el diodo schottky elegido para este proyecto es el modelo 10A05, el cual, atendiendo a su *datasheet* **[75]**, soporta hasta 10 amperios, 50 V_{DC} y 35 V_{RMS}.

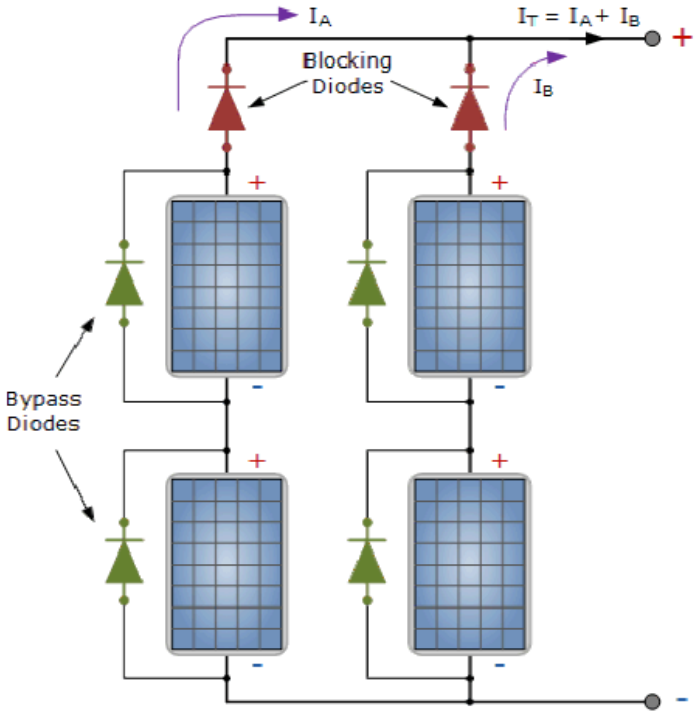


Figura 46: Ubicación de los diodos de *bypass* y de bloqueo

Teniendo en cuenta el amperaje máximo que se puede conseguir con estas celdas, se deben colocar cables de 1,5 mm^2 de sección. La instalación de este tipo de diodos se hará mediante el sistema de regletas y se incluirán conectores azules específicos para este tipo de cables. Como curiosidad, la palabra DIODO procede del griego DI-ODOS que significa dos caminos.

4.4.4.- Regulador de carga

El regulador de carga es un dispositivo electrónico que evita que la batería se sobrecargue, deteniendo la carga una vez esta se haya completado. Otra de sus funciones consiste en garantizar una carga constante preservando la vida de la batería. Este dispositivo es también un regulador de tensión. (Véase el siguiente apartado: *Regulador de tensión*).

Figura 47: Regulador de carga CTK5S en su versión de 6/12 V

Teniendo en cuenta los voltajes con los que se trabaja en este proyecto, el regulador de carga que se utilizará es el CTK5S que admite tensiones de entre 6 y 12 voltios y hasta 10 amperios. Existe una versión superior que eleva el voltaje a 24 V y el amperaje a 30 A.

En la parte central se conecta la batería, a la izquierda los cables provenientes de los paneles solares y a la derecha el sistema de alimentación. El regulador de carga es uno de los dispositivos más importantes y a su vez económicos del circuito.

4.4.5.- Regulador de tensión

Un regulador de tensión o conversor DC-DC es un dispositivo electrónico capaz de proporcionar una tensión a la salida diferente a la proporcionada a la entrada. Esto se puede conseguir mediante divisores de tensión u otros métodos.

Este proyecto trabajará con placas Arduino Mega 2560 y sensores HC-SR04 ambos alimentados a 5 voltios, módulos ESP32 a 3,3V y pantallas ILI9163C a 5V con retroiluminación a 3,3V. Además, se instalarán unas farolas a 3V con fines puramente estéticos. Por tanto, será necesario bajar la tensión de 6V a 5, 3,3 y 3 voltios cubriendo así las necesidades del proyecto.

De este modo, se incluirán dos reguladores de tensión LM2596 **[76]**. Se ha optado por este tipo de dispositivos a diferencia de los populares 78xx para tensiones positivas ó 79xx para tensiones negativas, porque a diferencia de estos últimos, no tienen tensión fija y disipan mejor el calor, además de soportar una corriente de hasta 3 amperios, lo cual obliga a utilizar cables de 1,5 mm^2 de sección.

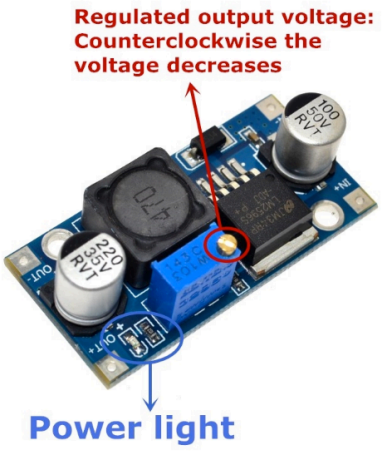


Figura 48: Regulador de tensión LM2596

Estos reguladores permiten modificar su tensión de salida mediante un pequeño tornillo de bronce situado en la parte central de la imagen con una horquilla que oscila entre los 1,2 y los 37 voltios ± 4%.

4.4.6.- Batería

Para un circuito alimentado por placas solares, se ha elegido una batería AGM de ciclo profundo **[77]**. Este tipo de baterías permiten cargas y descargas continuadas sin estropearse y no requieren de mantenimiento. Se trata de una batería económica de 6V con una capacidad de carga y descarga de 12 AH, siendo la versión encontrada inmediatamente superior a los requisitos de este proyecto.

Figura 49: Batería RT6120

4.4.7.- Pantallas

En este proyecto se utilizarán 7 pantallas ILI9163C **[78]**. Son pantallas TFT a color de 1,44” retroiluminadas con 4 diodos led. Utilizan el protocolo de comunicación SPI descrito en el apartado 4.4.3.2 de este proyecto.

Figura 50: Pantalla TFT a color de 1,44”

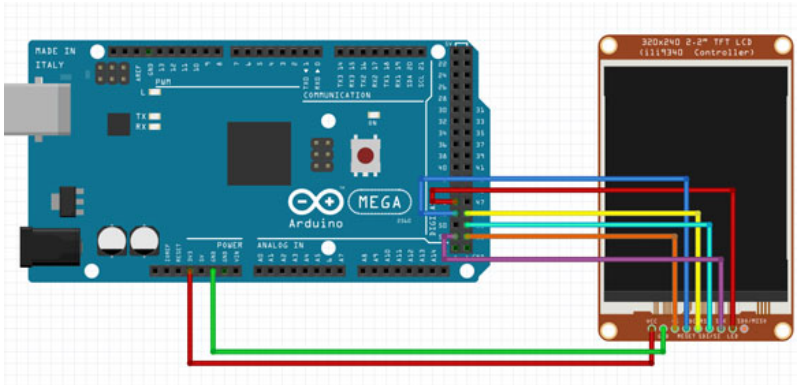


Figura 51: Conexionado de una pantalla en un Arduino Mega 2560

Para conseguir un encapsulado mucho más plano, se han desoldado los 8 pines de algunas de estas pantallas y soldado de nuevo por el otro lado de la pantalla.

Figura 52: Pantalla encapsulada en madera

Para programar estas pantallas será necesario utilizar las siguientes tres librerías:

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <TFT_ILI9163C.h>
```

Figura 53: Librerías necesarias para el uso de las pantallas TFT

Como cada pantalla dispone de 8 pines y habrá hasta 7 pantallas, el número total de cables necesarios sólo para estas pantallas se eleva a 56. Para simplificar este circuito se comparten los cables de los pines LED, SCK, SDA, GND y VCC reduciendo este valor a 5 cables comunes y 3 dedicados a cada pantalla.

4.4.8.- Otros componentes

a) Sensor LDR

Para la validación de este proyecto se incluirán varias farolas led a 3V con fines exclusivamente decorativos. Estas farolas se encenderán automáticamente mediante un sensor fotoeléctrico LDR (*Light Dependant Resistor* o Resistor Dependiente de la Luz, por sus siglas en inglés). Se trata de un resistor que modifica su valor cuando cambia la intensidad de la luz que incide en él **[79]**. Existe otro tipo de sensores fotoeléctricos como por ejemplo el BH1750 **[80]**.

b) Sensor FSR

El sistema de Smart Parking, objeto de este proyecto, es aplicable a muchos otros proyectos, como por ejemplo, el sistema de recogida de residuos urbanos mediante el uso de

sensores de fuerza resistivos FSR **[81]**, que permitan conocer el estado de los contenedores para priorizar la recogida en aquellos que se encuentren más llenos. En este proyecto se hará una simulación con tres sensores FSR 402.

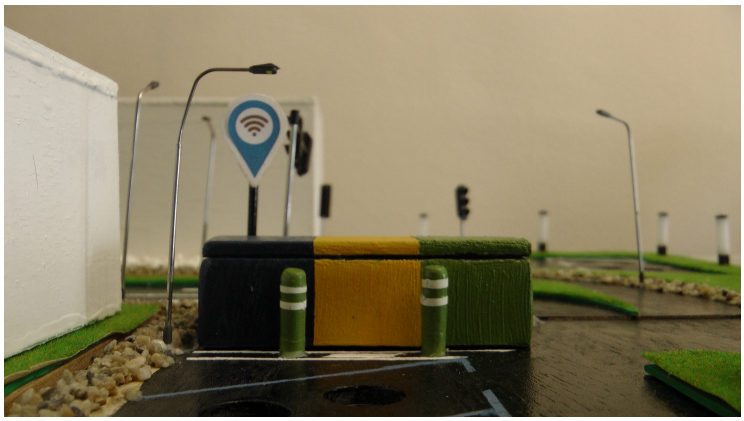


Figura 54: Zona de reciclaje

c) Transistor BJT

Otra aplicación podría ser el sistema de semáforos que pueden ser regulados con transistores BJT. De este modo, por ejemplo, los autobuses podrían modificar el estado de los semáforos, reduciendo los tiempos del transporte público.

4.4.9.- Pseudocódigos

A continuación se muestra el pseudocódigo que detalla la actuación de cada actor interviniente:

PSEUDOCÓDIGO ARDUINO EMISOR

```
Inicializa una cadena
Bucle de configuración
{
        Inicializa pines TRIGGER como salida
        Inicializa pines ECHO como entrada
        Inicializa comunicación serie (UART) con ESP32
}
Bucle de ejecución
{
        Inicializa una cadena auxiliar
        Desde el sensor 1 hasta el último
        {
                Envía un pulso ultrasónico
                Calcula el tiempo que tarda en volver
                Calcula la distancia con el objeto
                Si la distancia es mayor a 10 cm
                {
                        Escribe un 1 al final de la cadena auxiliar
                }
                Sino
                {
                        Escribe un 0 al final de la cadena auxiliar
                }
        }
        Si la cadena es distinta a la cadena auxiliar
        {
                Iguala cadena a cadena auxiliar
                Envía el resultado a ESP32 para su transmisión
        }
        Duerme 3 segundos
}
```

PSEUDOCÓDIGO ESP32 EMISOR

```
Incluye librería WiFi
Incluye librería específica para la comunicación con la plataforma
Declaración de constantes (SSID, contraseña, servidor, APIKEY, dispositivo, puerto y host)
Declaración de un cliente WiFi
Bucle de configuración
{
        Inicializa comunicación serie (UART) con Arduino Mega 2560
        Inicializa una conexión WiFi con el router
        Si la conexión no es satisfactoria
        {
                Espera 5 segundos
                Vuelve a intentarlo
        }
}
Bucle de ejecución
{
        Si se ha recibido una trama y la conexión es correcta
        {
                Envíala a la plataforma
        }
        Duerme hasta la siguiente trama recibida
}
```

PSEUDOCÓDIGO ESP32 RECEPTOR

```
Incluye librería WiFi
Incluye librería específica para la comunicación con la plataforma
Declaración de constantes (SSID, contraseña, servidor, APIKEY, dispositivo, puerto y host)
Declaración de un cliente WiFi
Bucle de configuración
{
        Inicializa comunicación serie (UART) con Arduino Mega 2560
        Inicializa una conexión WiFi con el router
        Si la conexión no es satisfactoria
        {
                Espera 5 segundos
                Vuelve a intentarlo
        }
}
Bucle de ejecución
{
        Si se ha recibido una trama y la conexión es correcta
        {
                Envíala al Arduino Mega 2560 receptor
        }
        Duerme hasta la siguiente trama recibida
}
```

PSEUDOCÓDIGO ARDUINO RECEPTOR

```
Incluye librería para el protocolo de comunicaciones SPI
Incluye librerías para el control de las pantallas ILI9163C
Define pines de conexión para las pantallas
Define todas las pantallas
Declara variables globales
Bucle de configuración
{
        Inicializa pines para el control de los semáforos como salida
        Pon todos los semáforos en rojo
        Inicializa comunicación serie (UART) con ESP32
        Inicializa todas las pantallas
        Rota todas las pantallas al modo 2
}
Bucle de ejecución
{
        Si se ha recibido una trama
        {
                Compara la última trama con la trama anterior
                Actualiza los valores que hayan sido modificados
                Iguala la trama anterior a la nueva trama recibida
        }
        Si ha pasado el tiempo suficiente
        {
                Actualiza el color de los semáforos
        }
        Duerme hasta recibir otra trama o haya pasado el tiempo para cambiar el color de los
semáforos
}
```

Capítulo 5

Escenario de validación

5.1.- Maqueta

5.1.1.- Escenario del demostrador

El escenario en el que se validará la tecnología desarrollada será una maqueta de 150 x 90 cm formada por 4 módulos periféricos de 60 x 30 cm, un módulo central de 30 x 30 cm y dos paneles solares de 90 x 30 cm situados a ambos lados de la maqueta. La estructura que presenta este entramado es la siguiente:

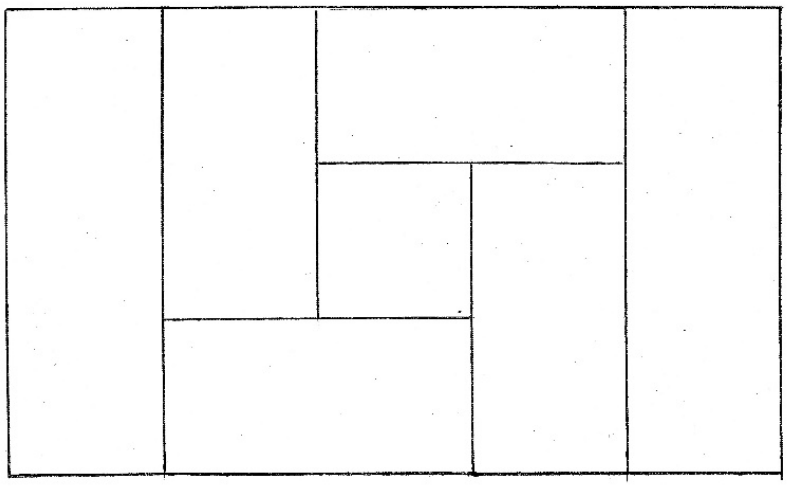


Figura 55: Estructura de la maqueta de validación

Esta maqueta contiene varios cruces especialmente diseñados para forzar la toma de decisiones por parte del conductor en base a la información ofrecida por los paneles estratégicamente situados en cada uno de estos cruces. Cuenta con 6 zonas de aparcamientos, cada una de las cuales tendrá asociada al menos una pantalla informativa.

Figura 56: Zonas de aparcamientos y pantallas asociadas

Los sensores situados en cada una de las plazas de estacionamiento serán los encargados de detectar la presencia de un vehículo, valor que será almacenado en la placa emisora y codificado para su posterior envío, en caso de haber sufrido alguna modificación, con el fin de actualizar la tabla de ocupación general con la situación de cada aparcamiento.

La placa receptora será la encargada de decodificar cualquier paquete que le llegue y actualizar aquellas pantallas que hayan sufrido algún cambio. Cada zona ha sido dimensionada para albergar un número de plazas suficientes para demostrar esta tecnología, con la única limitación del número de pines admitidos por el Arduino Mega 2560, haciendo uso de dos pines por sensor ultrasónico. La distribución de las plazas se muestra en la siguiente tabla:

Módulo	Zona	Plazas	Pantalla
1	1	7	1
1	2	4	2 y 3
2	3	5	4
2	4	5	5
3	5	3	6
4	6	2	7

Tabla 14: Distribución de las plazas de estacionamiento

5.1.2.- Identificación de las plazas de estacionamiento

Cada sensor deberá ir asociado a una única plaza y conectado a unos pines determinados de la placa. Concretamente se han elegido los siguientes:

Figura 57: Numeración de las 26 plazas en la maqueta

Número	1	2	3	4	5	6	7	8	9	10	11	12	13
Trigger	38	36	34	32	30	28	29	31	33	35	37	14	15
Echo	41	43	45	47	49	48	46	44	42	40	39	9	10

Número	14	15	16	17	18	19	20	21	22	23	24	25	26
Trigger	16	17	18	3	2	21	20	19	27	25	23	52	53
Echo	11	12	13	8	7	6	5	4	22	24	26	51	50

Tabla 15: Conexionado de los 26 sensores a los pines de la placa

5.1.3.- Experimentos sobre la maqueta

Para ahorrar costes en la validación de este proyecto se ha construido una maqueta con todas las posibilidades que ofrece el sistema. Gracias a ella se han podido corregir múltiples errores mediante el método de ensayo y error. Con la implementación de esta maqueta se han podido validar, entre otras, las siguientes afirmaciones:

- Correcto funcionamiento del sistema eléctrico.
- Correcto funcionamiento del código programado.
- Correcta comunicación entre los nodos de la maqueta y la plataforma web.
- Correcto funcionamiento de los sensores de la maqueta.
- Se han simulado casos reales y demostrado que es económicamente viable.
- Se ha demostrado que este sistema puede ser energéticamente autosuficiente.
- Se ha ampliado el uso de este sistema a la recogida de residuos urbanos.

Los falsos positivos en la maqueta son difíciles de abordar. En ocasiones, la mala conexión de un cable puede ocasionar el fallo de un sensor, dando lugar a un falso positivo. De hecho se tuvo que rediseñar todo el cableado para reducir este tipo de imprevistos.

Por otro lado, los sensores utilizados no incorporan ningún sistema de detección de errores, sin embargo, se puede sospechar de un falso positivo atendiendo a la distancia con el objeto. Si esta distancia se encuentra fuera del intervalo estándar de la altura de un vehículo promedio, debe sospecharse de un falso positivo. Al mismo tiempo, si la distancia es 0, el sensor no está funcionando correctamente, producido por una mala conexión del circuito o fallo de algún componente.

5.2.- Caso real

5.2.1.- Instalación de los equipos

Para un caso real en interiores, los sensores ultrasónicos deberán encapsularse en una carcasa de PVC e ir sujetos al techo. De ellos saldrá un tubo del mismo material con sección suficiente para dar cabida a cuatro cables de 0,25 mm^2.

En el caso de una instalación en exteriores, deberán utilizarse otro tipo de sensores como los magnetorresistivos, ya que los sensores ultrasónicos utilizados en este proyecto son sensibles a las condiciones meteorológicas. Bastará con anclar estos sensores al suelo mediante cuatro tornillos. Dependiendo de sus dimensiones, será necesario realizar una pequeña cavidad en el asfalto para albergarlos. Las puertas de enlace donde se conectarán los sensores más cercanos, estarán ancladas a diferentes farolas a una altura suficiente como para proporcionar una buena señal, así como para evitar cualquier acto vandálico. Por último, los paneles informativos deberán instalarse en torres independientes situadas en los laterales de la calzada.

5.2.2.- Experimentos sobre el terreno

Para validar este proyecto en un caso real se han llevado a cabo una serie de ensayos en un parking público al aire libre con sensores ultrasónicos HC-SR04.

Figura 58: Parking donde se han llevado a cabo las pruebas

En primer lugar se ha conectado uno de estos sensores al nodo de monitorización mediante cuatro cables de un metro y medio de largo cada uno. Este sensor se ha dispuesto en el centro de una de las plazas de estacionamiento donde se realizaron las pruebas.

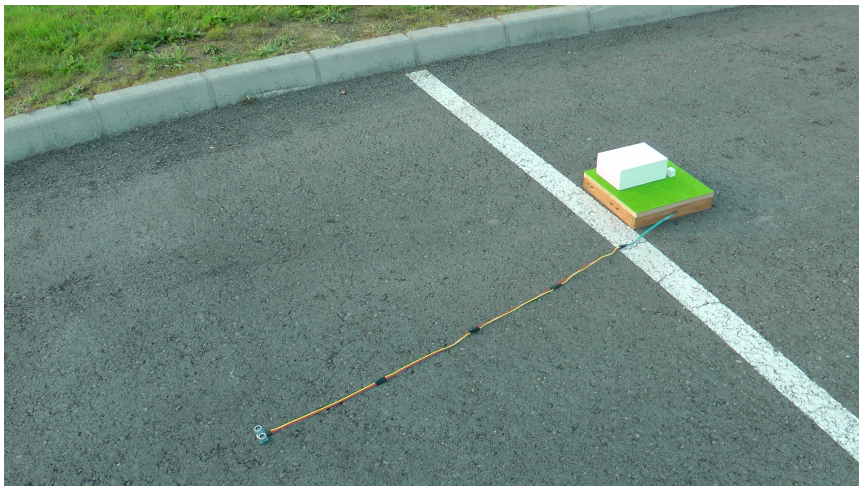


Figura 59: Sensor conectado al nodo de monitorización en una de las plazas

En un principio se planteó esta distribución, pero se ha observado que los cables sufrían cada vez que un vehículo estacionaba al pasar por encima de ellos, por lo que se valoraron otras alternativas. Además, la longitud de los cables era demasiado corta ya que el nodo de distribución debería situarse más lejos, sin invadir otras plazas.

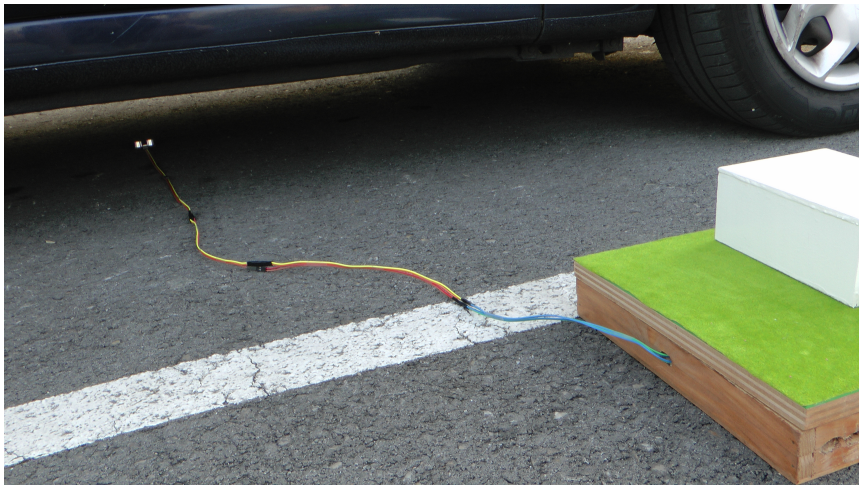


Figura 60: Vehículo estacionado con un sensor detectando su presencia

El siguiente problema vino dado por la carencia de conectividad a la red en esa zona, por lo que una vez puesto en funcionamiento el sistema, este no disponía de una red wifi próxima. Para solventarlo se creó una red wifi con un dispositivo móvil configurado como Punto de Acceso Inalámbrico compartiendo la tarifa de datos del terminal, algo conocido como *tethering* **[82]**. En un caso real deberán instalarse routers anclados a farolas o torres cercanas que proporcionen acceso inalámbrico a internet a todos los nodos conectados.

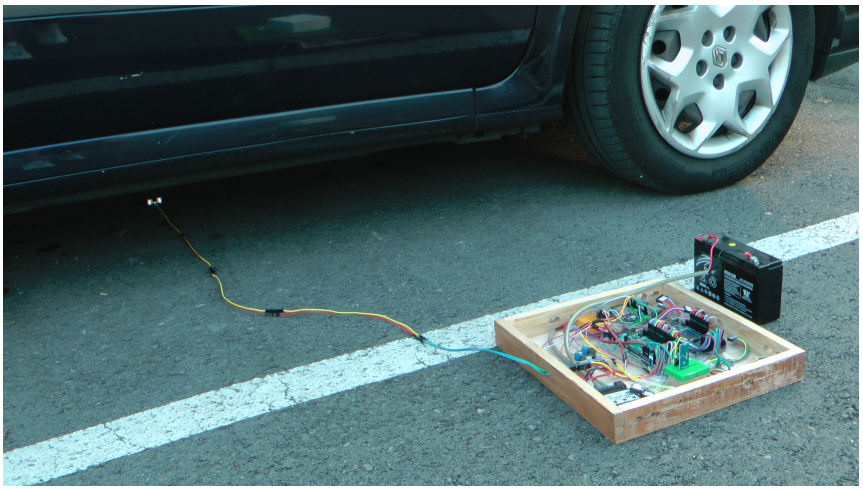


Figura 61: Nodo de monitorización conectado a internet mediante una red wifi compartida

Para evitar pisar los cables, se plantearon otras distribuciones como la frontal como la más económica. Actualmente estos sensores se están colocando en el techo en los parkings públicos de pago, evitando no solo este problema, sino otros que pudieran dañar el sensor como gotas de aceite, suciedad o que el vehículo le transcurra por encima o lo golpee de algún modo. Evidentemente la obstrucción del sensor, ya sea por un objeto, una persona o incluso el propio encapsulado del sensor, hará que deje de funcionar correctamente.

Otro inconveniente reside en el umbral de detección establecido en el código del Arduino emisor, el cual es de 10 cm para la maqueta, y se ha tenido que ampliar a 50 cm para este experimento. Según el fabricante, la distancia máxima a la que operan estos sensores puede alcanzar los 4 metros, por lo que la nueva distancia establecida se encuentra dentro del rango de operatividad del dispositivo.

Figura 62: Distribución frontal del sensor

Otras distribuciones pudieran ser peraltando el sensor o integrándolo directamente dentro o sobre el bordillo en cuyo caso el cableado necesario sería menor.

Figura 63: Distribución frontal con sensor peraltado

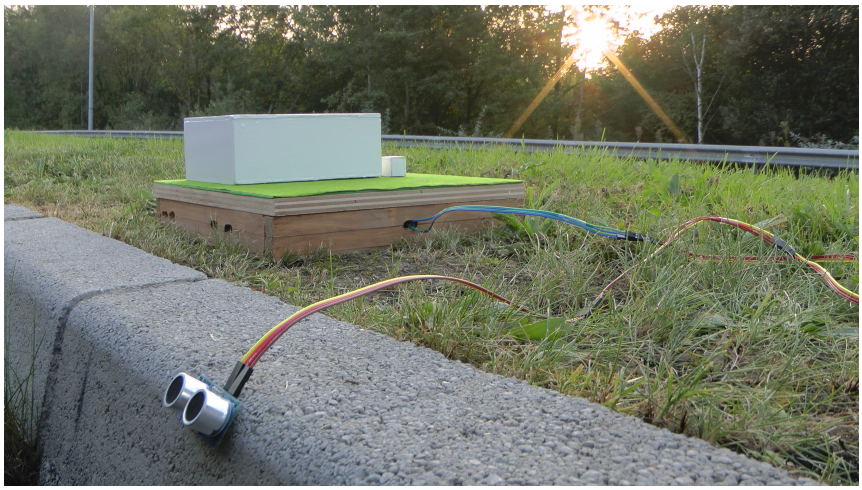


Figura 64: Distribución frontal con sensor dentro o sobre el bordillo

Finalmente, también se deben tener en cuenta los problemas meteorológicos como la lluvia o el viento. El uso de ultrasonidos implica no poder cubrir los transductores del sensor, los cuales son altamente sensibles al agua. Además, es fundamental evitar el aire en la transmisión puesto que una capa de este gas podría anular la propagación de la onda ultrasónica, dada la alta atenuación que proporciona **[83]**. Por estos motivos se desaconseja el uso de esta tecnología en exteriores tal y como ya se comentó en el apartado anterior.

5.2.3.- Montaje en un parking cubierto

En el caso de un parking cubierto se tiene la ventaja de que todos los sensores pueden instalarse en el techo. A continuación se muestra la instalación de un caso real en funcionamiento correspondiente a un parking público de pago en la ciudad de Valencia.

Figura 65: Led indicativo de plaza ocupada (izquierda) y sensor encapsulado (derecha)

Figura 66: Montaje completo

Capítulo 6

Conclusiones

Como ya se ha visto en las pruebas de validación de un caso real, este proyecto no sería viable en un área sin cobertura WiFi o 3G/4G que permita dar salida a internet a los paquetes de datos con la información recogida por todos los sensores del sistema. También se ha mencionado previamente que los sensores ultrasónicos utilizados en esta simulación no son útiles en exteriores ya que esta tecnología es muy sensible a las condiciones meteorológicas. En su defecto deberán instalarse otro tipo de sensores como pudieran ser los sensores magnetorresistivos, tal y como se explica en el punto 3.4 de este proyecto.

La distancia máxima entre nodos dependerá de la versión WiFi utilizada y de la potencia de las antenas de los dispositivos, además de los posibles obstáculos que se encuentren e interferencias con otros sistemas inalámbricos. Para alimentar los nodos de interconexión, se pueden instalar paneles fotovoltaicos que proporcionen energía suficiente para su correcto funcionamiento, tal y como se demuestra en la maqueta de este proyecto.

Algunos de los múltiples inconvenientes que se han ido encontrando a lo largo de este proyecto y que se han ido tratando son:

- Era necesario reducir el tiempo que tardan los datos en actualizarse. Fue necesario añadir retardos en el código porque estaba programado para enviar tramas de datos a la plataforma cada vez que hubiera alguna modificación pero a veces fallan algunos sensores y envía demasiadas tramas alcanzando el límite de la versión gratuita que ofrece *Carriots*.
- Hubo que colocar resistencias de 1 y 2 K para hacer un divisor de tensión para conectar el pin Tx del Arduino con el pin Rx del ESP32 porque el Arduino Mega funciona a 5V y el ESP32 va a 3,3V.
- Al encender o apagar las farolas de la maqueta, las pantallas modificaban su luminosidad ya que estaban conectadas al mismo regulador de tensión que las farolas. Para evitar este inconveniente se tuvo que instalar un tercer regulador de tensión separando así el circuito de las farolas del de las pantallas.
- Hubo que colocar varias tomas a tierra porque el cable era demasiado delgado para suministrar toda la corriente.
- Este proyecto se podría haber hecho con una sola antena y un solo Arduino. O incluso se podría haber hecho una versión reducida con una única antena ESP32 y sin ningún Arduino.
- Fue necesario descargarse varias librerías para manejar las pantallas y para “dormir” las placas.
- Fue necesario instalar los *drivers* para programar las placas ESP32 en el software de Arduino ya que no vienen por defecto.

- Fue necesario reestructurar todos los cables canalizándolos y numerándolos ya que conforme iba creciendo el proyecto se hacía imposible distinguirlos. Se utilizaron más de 150 metros de cable en la maqueta.
- Fue necesario modificar todas las uniones fortaleciéndolas con estaño y tubos termorretráctiles.
- Algunas plazas no funcionaban, por lo que fue necesario buscar dónde fallaban las conexiones con un óhmetro.
- Conecté las placas ESP32 al pin de 3,3V de uno de los Arduinos pero no funcionaban. Posteriormente averigüé que este pin no proporciona la corriente suficiente para alimentarlos. Tan sólo proporciona 50 mA de los 520 mA necesarios, por lo que tuve que conectarlos a otra toma. Su consumo es tan elevado que suelen requerir de alimentación externa en función de la proximidad al punto de acceso inalámbrico.
- He tenido varios problemas de conexión que resolví mediante el rastreo de tramas con el software *WireShark*, el cual me permitió descubrir que uno de los problemas se debía a una desconexión del servidor por un certificado SSL no válido.

Capítulo 7

Referencias bibliográficas

Referencias bibliográficas

[1] - http://www.europapress.es/madrid/noticia-transporte-genera-41-emisiones-contaminantes-madrid-estudio-20170711142607.html

[2] - http://www.expansion.com/economia-digital/innovacion/2017/02/11/589de1d7ca4741ac518b461e.html

[3] - http://www.asesga.org/documentos/revista_aparcar/Aparcar_40.pdf

[4] - http://www.azuqueca.es/fileadmin/azuqueca/urbanismo/Estudio_movilidad/Sintesis_estudio_movilidad.ppt

[5] - http://bicihome.com/el-coste-economico-de-los-atascos-en-coche/

[6] - http://www.endesaeduca.com/Endesa_educa/recursos-interactivos/smart-city/

[7] - http://www.antena3.com/noticias/tecnologia/2020-numero-dispositivos-conectados-internet-triplicara-poblacion-mundial_20161128583c6e560cf264101b1be653.html

[8] - http://www.ipv6.es/es-ES/introduccion/Paginas/QueesIPv6.aspx

[9] - https://www.by.com.es/blog/que-es-rfid/

[10] - http://www.zemsania.com/smart-mobility-la-clave-para-una-mejor-movilidad-sostenible/

[11] - http://www.triplepundit.com/2016/01/zero-emissions-smart-mobility-city/

[12] - https://web.ua.es/es/smart/smart-environment-un-entorno-de-calidad-de-vida.html

[13] - Libro: “Redes inalámbricas de sensores: teoría y aplicación práctica”.
Roberto Fernández Martínez, Joaquín Ordieres Meré, Francisco Javier Martínez de Pisón Ascacíbar, Ana González Marcos, Fernando Alba Elías, Rubén Lostado Lorza, Alpha Verónica Pernía Espinoza e Integrantes del grupo de investigación EDMANS.
Universidad de la Rioja. ISBN 978-84-692-3007-7. Año 2009.

[14] - http://standards.ieee.org/about/get/802/802.11.html

[15] - IEEE 802.11. Standard for Telecommunications and Information Exchange Between Systems Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Nueva York. The IEEE, Inc. Año 1999.

[16] - Libro: "Mobile Computing. Technology, Applications and Service Creation".
Asoke K Talukder y Roopa R Yavagal. ISBN 978-0-07-058807-3. Año 2005.

[17] - http://standards.ieee.org/getieee802/download/802.11b-1999.pdf

[18] - http://standards.ieee.org/getieee802/download/802.11g-2003.pdf

[19] - http://standards.ieee.org/getieee802/download/802.11n-2009.pdf

[20] - http://standards.ieee.org/getieee802/download/802.11ac-2013.pdf

[21] - https://www.xatakamovil.com/conectividad/wifi-ac-ya-no-es-el-mas-rapido-la-era-del-wifi-ad-ha-comenzado-y-estas-son-sus-ventajas

[22] - https://norfipc.com/redes/tipos-redes-estandares-wi-fi-diferencias.php

[23] - http://plcwifi.es/wifi-ac/

[24] - https://www.maestrodelacomputacion.net/diferencias-entre-wep-wpa-wpa2-aes-tkip/

[25] - https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

[26] - http://standards.ieee.org/about/get/802/802.3.html

[27] - Libro: "Análisis de redes y sistemas de comunicaciones".
Escrito por Xavier Hesselbach Serra y Jordi Altés Bosch (2002). Edicions UPC. ISBN 84-8301-611-7.

[28] - http://mirelucx.over-blog.com/article-30722153.html

[29] - http://es.ccm.net/contents/672-ethernet

[30] - http://www.cisco.com/c/en/us/support/docs/routers/10000-series-routers/46792-ethbase.html

[31] - https://www.pilz.com/es-ES/knowhow/lexicon/articles/073180

[32] - Libro: "Information Security Management Handbook".
Harold F. Tipton y Micki Krause. ISBN 1-4200-1358-0

[33] - https://kb.wisc.edu/ns/page.php?id=7829

[34] - http://www.thenetworkencyclopedia.com/entry/1000baselx

[35] - http://www.thenetworkencyclopedia.com/entry/1000basesx

[36] - http://www.thenetworkencyclopedia.com/entry/1000basecx

[37] - http://www.thenetworkencyclopedia.com/entry/1000baset

[38] - http://standards.ieee.org/getieee802/download/802.3ae-2002.pdf

[39] - Libro: "Webster's New World Telecom Dictionary"
Escrito por Ray Horak. ISBN 978-0-471-77457-0

[40] - http://www.ieee802.org/3/bs/

[41] - http://www.ieee802.org/15/

[42] - http://www.ieee802.org/16/

[43] - http://www.mibqyyo.com/articulos/2014/12/30/lte-o-wimax/#/vanilla/discussion/embed/?vanilla_discussion_id=0

[44] - https://www.xataka.com/otros/lte-y-lte-advanced-cual-de-ellos-es-realmente-4g

[45] - https://www.xataka.com/moviles/5g-asi-es-el-futuro-de-las-redes-moviles

[46] - https://www.alertlogic.com/blog/where-is-ipv1,-2,-3,and-5/

[47] - Cerf, V.; Kahn, R. (1974). «A Protocol for Packet Network Intercommunication». IEEE Transactions on Communications. COM-22 (5): 637-648.

[48] - https://www.arduino.cc/en/Reference/Serial

[49] - http://microcontroladores-mrelberni.com/usart-pic-comunicacion-serial/

[50] - https://www.arduino.cc/en/Reference/SPI

[51] - http://www.mct.net/faq/spi.html

[52] - https://aprendiendoarduino.wordpress.com/2016/11/06/icsp/

[53] - http://www.prometec.net/bus-i2c/

[54] - http://www.diarioelectronicohoy.com/blog/introduccion-al-i2c-bus

[55] - https://www.carriots.com/

[56] - http://reportedigital.com/iot/plataformas-iot-mercado-tecnologico/

[57] - http://soloelectronicos.com/2015/02/22/50-proveedores-de-iot/

[58] - https://aws.amazon.com/es/iot-platform/

[59] - https://www.ibm.com/es-es/marketplace/internet-of-things-cloud

[60] - https://cloud.google.com/solutions/iot/?hl=es

[61] - http://www.zatar.com/blog/what-is-an-iot-application-platform

[62] - https://secmotic.com/blog/plataforma-iot/

[63] - Libro: "Ingeniería del software y bases de datos: tendencias actuales".
Escrito por Isidro Ramos Salavert y María Dolores Lozano Pérez (2000). Universidad de Castilla La Mancha. ISBN 8484270777. Capítulo 6.4: Entornos de Desarrollo Integrados, pág. 78.

[64] - https://www.arduino.cc/en/Guide/Environment

[65] - https://aprendiendoarduino.wordpress.com/2017/06/19/que-es-arduino-y-hardware-libre-2/

[66] - Libro: "Arduino Notebook: A Beginner's Reference"
Escrito por Brian W. Evans.

[67] - http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

[68] - http://todohard.awardspace.com/tutrs232/rs232-uart-1.htm

[69] - https://store.arduino.cc/arduino-mega-2560-rev3

[70] - http://www.espressif.com/en/products/hardware/esp32/overview

[71] - http://blog.bricogeek.com/noticias/electronica/comparativa-y-analisis-completo-de-los-modulos-wifi-esp8266-y-esp32/

[72] - http://www.nbsolar.com

[73] - http://www.omnisun.it/wp-content/uploads/2013/04/Sun-Earth-Solar-125x125-Mono-Cell-Parameters.pdf

[74] - http://www.grupoelektra.es/blog/wp-content/uploads/2014/09/como-somos-los-delektra-bypass-y-bloqueo.pdf

[75] - http://pdf1.alldatasheet.es/datasheet-pdf/view/190269/WTE/10A10/+0J1_JAVTSEa88EAA+/datasheet.pdf

[76] - http://www.ti.com/lit/ds/symlink/lm2596.pdf

[77] - http://www.mantech.co.za/Datasheets/Products/RT6120.pdf

[78] - http://www.buydisplay.com/download/ic/ILI9163.pdf

[79] - https://reviseomatic.org/help/e-resistors/Resistors%20-%20Light%20Dependent.php

[80] - http://rohmfs.rohm.com/en/products/databook/datasheet/ic/sensor/light/bh1750fvi-e.pdf

[81] - www.trossenrobotics.com/productdocs/2010-10-26-DataSheet-FSR402-Layout2.pdf

[82] - https://andro4all.com/2013/12/tethering-android

[83] - https://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_03_04/infra_y_ultra/propagacion_ultrasonidos.htm

Capítulo 8

Anexos

8.1.- Códigos fuente

8.1.1.- Código fuente Arduino emisor

```
#include "LowPower.h"

int echo[26]    = {49,48,46,44,42,40,39,    41,43,45,47,
4,5,6,7,8,      9,10,11,12,13,     22,24,26,   51,50};
int trigger[26] = {30,28,29,31,33,35,37,    38,36,34,32,
19,20,21,3,2,   14,15,16,17,18,    27,25,23,   52,53};
int secuencia[30];

void setup() {
  /* ECHO */                            /* TRIGGER */
  // Zona azul
  pinMode(4, INPUT);                    pinMode(14, OUTPUT);
  pinMode(5, INPUT);                    pinMode(15, OUTPUT);
  pinMode(6, INPUT);                    pinMode(16, OUTPUT);
  pinMode(7, INPUT);                    pinMode(17, OUTPUT);
  pinMode(8, INPUT);                    pinMode(18, OUTPUT);
  pinMode(9, INPUT);                    pinMode(19, OUTPUT);
  pinMode(10, INPUT);                   pinMode(20, OUTPUT);
  pinMode(11, INPUT);                   pinMode(21, OUTPUT);
  pinMode(12, INPUT);                   pinMode(2, OUTPUT);
  pinMode(13, INPUT);                   pinMode(3, OUTPUT);

  // Centro comercial
  pinMode(39, INPUT);                   pinMode(28, OUTPUT);
  pinMode(40, INPUT);                   pinMode(29, OUTPUT);
  pinMode(41, INPUT);                   pinMode(30, OUTPUT);
  pinMode(42, INPUT);                   pinMode(31, OUTPUT);
  pinMode(43, INPUT);                   pinMode(32, OUTPUT);
  pinMode(44, INPUT);                   pinMode(33, OUTPUT);
  pinMode(45, INPUT);                   pinMode(34, OUTPUT);
  pinMode(46, INPUT);                   pinMode(35, OUTPUT);
  pinMode(47, INPUT);                   pinMode(36, OUTPUT);
  pinMode(48, INPUT);                   pinMode(37, OUTPUT);
  pinMode(49, INPUT);                   pinMode(38, OUTPUT);

  // Hotel
  pinMode(22, INPUT);                   pinMode(25, OUTPUT);
  pinMode(26, INPUT);                   pinMode(23, OUTPUT);
  pinMode(24, INPUT);                   pinMode(27, OUTPUT);

  // Parking privado
  pinMode(50, INPUT);                   pinMode(52, OUTPUT);
  pinMode(51, INPUT);                   pinMode(53, OUTPUT);

  // Farolas
  pinMode(A10,INPUT);

  // Comunicación serie con ESP32
  Serial.begin(115200);
}

void loop() {
  long tiempo, distancia;
```

```
  int secuencia_aux[30], distinto = false;

  for(int i=0; i<26; i++){
    // Estabiliza el sensor
    digitalWrite(trigger[i],LOW);
    delayMicroseconds(5);

    // Envía el pulso ultrasónico
    digitalWrite(trigger[i], HIGH);
    delayMicroseconds(5);

    // Calcula el tiempo que tarda en volver
    tiempo = pulseIn(echo[i], HIGH);

    // Calcula la distancia con el objeto
    distancia = int(0.017*tiempo);

    if(distancia > 10)
      secuencia_aux[i] = true;
    else
      secuencia_aux[i] = false;
  }

  secuencia_aux[26] = reciclaje(12);
  secuencia_aux[27] = reciclaje(13);
  secuencia_aux[28] = reciclaje(11);
  secuencia_aux[29] = farolas();

  for(int i=0; i<30; i++){
    if(secuencia_aux[i] != secuencia[i]){
      secuencia[i] = secuencia_aux[i];
      distinto = true;
    }
  }

  if(distinto){
    for(int i=0; i<30; i++)
      Serial.print(secuencia[i]);
  }

  // Duerme 4 segundos
  LowPower.powerDown(SLEEP_4S, ADC_OFF, BOD_OFF);
}

int reciclaje(int i){
  if (analogRead(i) > 100)
    return true;
  else
    return false;
}

bool farolas(){
  if (analogRead(A10) < 768)
    return false;
  else
    return true;
}
```

8.1.2.- Código fuente Arduino receptor

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <TFT_ILI9163C.h>
#include "LowPower.h"

// Definición de colores
#define BLACK   0x0000
#define BLUE    0x001F
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF

// Definición de pines
#define __CS1 26
#define __CS2 32
#define __CS3 33
#define __CS4 38
#define __CS5 39
#define __CS6 44
#define __CS7 45

#define __DC1 24 // Reset
#define __DC2 30
#define __DC3 31
#define __DC4 36
#define __DC5 37
#define __DC6 42
#define __DC7 43

#define __A01 22
#define __A02 28
#define __A03 29
#define __A04 34
#define __A05 35
#define __A06 40
#define __A07 41

// Declara las 7 pantallas
TFT_ILI9163C pantalla_1 = TFT_ILI9163C(__CS1, __A01, __DC1);
TFT_ILI9163C pantalla_2 = TFT_ILI9163C(__CS2, __A02, __DC2);
TFT_ILI9163C pantalla_3 = TFT_ILI9163C(__CS3, __A03, __DC3);
TFT_ILI9163C pantalla_4 = TFT_ILI9163C(__CS4, __A04, __DC4);
TFT_ILI9163C pantalla_5 = TFT_ILI9163C(__CS5, __A05, __DC5);
TFT_ILI9163C pantalla_6 = TFT_ILI9163C(__CS6, __A06, __DC6);
TFT_ILI9163C pantalla_7 = TFT_ILI9163C(__CS7, __A07, __DC7);

String trama[32];
int libres[6] = {0,0,0,0,0,0};
unsigned long tiempo = millis();
unsigned long tiempo2 = millis();
unsigned long tiempo3 = millis();
int secuencia = 1;
```

```
bool oferta = false;
bool H_hotel = false;
bool semaf = false;

void setup(void) {
  // Inicializa la comunicación serie con ESP32 receptor
  Serial.begin(115200);

  // SEMÁFOROS

  // Individual fuera del hotel
  pinMode(2, OUTPUT); // Verde
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);

  // Individual dentro del hotel
  pinMode(5, OUTPUT); // Verde
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

  // Doble para hotel
  pinMode(8, OUTPUT); // Verde
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);

  // Doble fuera
  pinMode(11, OUTPUT); // Verde
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);

  // Inicializa los semáforos
  semaforos(0);

  // DISPLAYS

  // Inicializa las 7 pantallas
  pantalla_1.begin();
  pantalla_2.begin();
  pantalla_3.begin();
  pantalla_4.begin();
  pantalla_5.begin();
  pantalla_6.begin();
  pantalla_7.begin();

  // Rota las 7 pantallas
  pantalla_1.setRotation(2);
  pantalla_2.setRotation(2);
  pantalla_3.setRotation(2);
  pantalla_4.setRotation(2);
  pantalla_5.setRotation(2);
  pantalla_6.setRotation(2);
  pantalla_7.setRotation(2);

  fondo(WHITE,1);
  pantalla_1.setCursor(10,50);
  pantalla_1.setTextSize(2);
  pantalla_1.setTextColor(BLACK);
```

```
  pantalla_1.print("Smart");
  pantalla_1.setTextColor(BLUE);
  pantalla_1.print("Blue");

  fondo(WHITE,2);
  pantalla_2.setCursor(10,50);
  pantalla_2.setTextSize(2);
  pantalla_2.setTextColor(BLACK);
  pantalla_2.print("Smart");
  pantalla_2.setTextColor(BLUE);
  pantalla_2.print("Blue");

  fondo(WHITE,3);
  pantalla_3.setCursor(10,50);
  pantalla_3.setTextSize(2);
  pantalla_3.setTextColor(BLACK);
  pantalla_3.print("Smart");
  pantalla_3.setTextColor(BLUE);
  pantalla_3.print("Blue");

  fondo(WHITE,4);
  pantalla_4.setCursor(10,50);
  pantalla_4.setTextSize(2);
  pantalla_4.setTextColor(BLACK);
  pantalla_4.print("Smart");
  pantalla_4.setTextColor(BLUE);
  pantalla_4.print("Blue");

  fondo(WHITE,5);
  pantalla_5.setCursor(10,50);
  pantalla_5.setTextSize(2);
  pantalla_5.setTextColor(BLACK);
  pantalla_5.print("Smart");
  pantalla_5.setTextColor(BLUE);
  pantalla_5.print("Blue");

  fondo(WHITE,6);
  pantalla_6.setCursor(10,50);
  pantalla_6.setTextSize(2);
  pantalla_6.setTextColor(BLACK);
  pantalla_6.print("Smart");
  pantalla_6.setTextColor(BLUE);
  pantalla_6.print("Blue");

  fondo(WHITE,7);
  pantalla_7.setCursor(10,50);
  pantalla_7.setTextSize(2);
  pantalla_7.setTextColor(BLACK);
  pantalla_7.print("Smart");
  pantalla_7.setTextColor(BLUE);
  pantalla_7.print("Blue");

  // Farolas
  pinMode(A0,OUTPUT);
}
```

```
void loop() {
  if (Serial.available() > 0){
    String trama_aux[32];
    bool distinto = false;

  // Recibe el estado de los sensores y lo almacena en trama_aux
    for(int i=0; i<32; i++){
      trama_aux[i] = Serial.read() - 48;
      Serial.print(trama_aux[i]);
      if(trama[i] != trama_aux[i]){
        trama[i] = trama_aux[i];
        distinto = true;
      }
    }
    Serial.println();

    if(distinto){
      actualizaPantallas();
      actualizaFarolas();
      actualizaSemaforos();
    }
  }

  // Semáforos. Si han transcurrido 20 seg. desde la última vez:
  if((millis() - tiempo) > 20000 && !semaf){
    tiempo = millis();
    semaforos(secuencia);
    secuencia++;
    if(secuencia == 4)
      secuencia = 1;
  }
  if(trama[31]=="5")
    semaforos(4);

  // Ofertas. Si han transcurrido 25 seg. desde la última vez:
  if((millis() - tiempo2) > 25000){
    if((millis() - tiempo2) > 35000){
      tiempo2 = millis();
      actualizaPantalla(0);
      oferta = false;
    }else{
      if(!oferta){
        ofertas();
        oferta = true;
      }
    }
  }

  // Hotel. Si han transcurrido 12 segundos desde la última vez:
  if((millis() - tiempo3) > 12000){
    if((millis() - tiempo3) > 16000){
      tiempo3 = millis();
      actualizaPantalla(4);
      H_hotel = false;
    }else{
      if(!H_hotel){
        hotel();
```

```
        H_hotel = true;
      }
    }
  }

  delay(100);

  // Duerme 4 segundos
   LowPower.powerDown(SLEEP_4S, ADC_OFF, BOD_OFF);
}

// Modifica la secuencia de los semáforos
void actualizaSemaforos(){
  switch(trama[31].toInt()){
  case 0:
  case 1:
  case 4:
    tiempo = millis();
    semaforos(0);
    secuencia = 1;
    semaf = false;
    break;
  case 2:
    tiempo = millis();
    secuencia = 2;
    semaf = false;
    semaforos(5);
    semaforo_2(1);
    break;
  case 3:
    tiempo = millis();
    secuencia = 3;
    semaf = false;
    semaforos(5);
    semaforo_3(1);
    break;
  case 5:
    semaforos(4);
    semaf = true;
    break;
  case 6:
    semaforos(5);
    semaf = true;
    break;
  default:
    break;
  }
}

// Enciende o apaga las farolas
void actualizaFarolas(){
  if(trama[30] == "0"){
    if(trama[29] == "1")
      digitalWrite(A0, HIGH);
    else
      digitalWrite(A0, LOW);
  }else{
```

```
    if(trama[30] == "1")
      digitalWrite(A0, HIGH);
    else
      digitalWrite(A0, LOW);
  }
}

// Actualiza todas las pantallas
void actualizaPantallas(){
  int libres_aux[6] = {0,0,0,0,0,0};
  for(int i=0; i<26; i++){
    if(trama[i] == "1"){
      if(i<7){
        libres_aux[0]++;
      }else if(i<11){
        libres_aux[1]++;
      }else if(i<16){
        libres_aux[2]++;
      }else if(i<21){
        libres_aux[3]++;
      }else if(i<24){
        libres_aux[4]++;
      }else{
        libres_aux[5]++;
      }
    }
  }
  // Asigna el valor a las pantallas
  for(int i=0; i<6; i++){
    if(libres[i] != libres_aux[i]){
      libres[i] = libres_aux[i];
      actualizaPantalla(i);
    }
  }
}

// Actualiza una pantalla
void actualizaPantalla(int i){
  switch(i){
    case 0:
      pantalla_1.setTextColor(BLACK);
      if(libres[i] > 0){
        pantalla_1.setCursor(50,25);
        pantalla_1.setTextSize(6);
        fondo(GREEN,1);
        pantalla_1.print(libres[i]);
        pantalla_1.setCursor(20,80);
        pantalla_1.setTextSize(3);
        pantalla_1.print("LIBRE");
      }else{
        pantalla_1.setCursor(15,40);
        pantalla_1.setTextSize(2);
        fondo(RED,1);
        pantalla_1.print("COMPLETO");
      }
      break;
    case 1:
```

```
      pantalla_2.setTextColor(BLACK);
      if(libres[i] > 0){
        pantalla_2.setCursor(50,25);
        pantalla_2.setTextSize(6);
        fondo(GREEN,2);
        pantalla_2.print(libres[i]);
        pantalla_2.setCursor(20,80);
        pantalla_2.setTextSize(3);
        pantalla_2.print("LIBRE");
      }else{
        pantalla_2.setCursor(15,40);
        pantalla_2.setTextSize(2);
        fondo(RED,2);
        pantalla_2.print("COMPLETO");
      }
      pantalla_3.setTextColor(BLACK);
      if(libres[i] > 0){
        pantalla_3.setCursor(50,25);
        pantalla_3.setTextSize(6);
        fondo(GREEN,3);
        pantalla_3.print(libres[i]);
        pantalla_3.setCursor(20,80);
        pantalla_3.setTextSize(3);
        pantalla_3.print("LIBRE");
      }else{
        pantalla_3.setCursor(15,40);
        pantalla_3.setTextSize(2);
        fondo(RED,3);
        pantalla_3.print("COMPLETO");
      }
      break;
    case 2:
      pantalla_4.setTextColor(BLACK);
      if(libres[i] > 0){
        pantalla_4.setCursor(50,25);
        pantalla_4.setTextSize(6);
        fondo(GREEN,4);
        pantalla_4.print(libres[i]);
        pantalla_4.setCursor(20,80);
        pantalla_4.setTextSize(3);
        pantalla_4.print("LIBRE");
      }else{
        pantalla_4.setCursor(15,40);
        pantalla_4.setTextSize(2);
        fondo(RED,4);
        pantalla_4.print("COMPLETO");
      }
      break;
    case 3:
      pantalla_5.setTextColor(BLACK);
      if(libres[i] > 0){
        pantalla_5.setCursor(50,25);
        pantalla_5.setTextSize(6);
        fondo(GREEN,5);
        pantalla_5.print(libres[i]);
        pantalla_5.setCursor(20,80);
        pantalla_5.setTextSize(3);
```

```
        pantalla_5.print("LIBRE");
      }else{
        pantalla_5.setCursor(15,40);
        pantalla_5.setTextSize(2);
        fondo(RED,5);
        pantalla_5.print("COMPLETO");
      }
      break;
    case 4:
      pantalla_6.setTextColor(BLACK);
      if(libres[i] > 0){
        pantalla_6.setCursor(50,25);
        pantalla_6.setTextSize(6);
        fondo(GREEN,6);
        pantalla_6.print(libres[i]);
        pantalla_6.setCursor(20,80);
        pantalla_6.setTextSize(3);
        pantalla_6.print("LIBRE");
      }else{
        pantalla_6.setCursor(15,40);
        pantalla_6.setTextSize(2);
        fondo(RED,6);
        pantalla_6.print("COMPLETO");
      }
      break;
    case 5:
      pantalla_7.setTextColor(BLACK);
      if(libres[i] > 0){
        pantalla_7.setCursor(50,25);
        pantalla_7.setTextSize(6);
        fondo(GREEN,7);
        pantalla_7.print(libres[i]);
        pantalla_7.setCursor(20,80);
        pantalla_7.setTextSize(3);
        pantalla_7.print("LIBRE");
      }else{
        pantalla_7.setCursor(15,40);
        pantalla_7.setTextSize(2);
        fondo(RED,7);
        pantalla_7.print("COMPLETO");
      }
      break;
    default:
      break;
  }
}

// Imprime un color de fondo en una pantalla
void fondo(uint16_t color, int pantalla){
  switch(pantalla){
    case 1:
      for (int16_t x=pantalla_1.height()-1; x > 25; x-=25)
        pantalla_1.fillRect((pantalla_1.width()-1)/2 -x/2,
(pantalla_1.height()-1)/2 -x/2 , x, x, color);
      break;
```

```
    case 2:
      for (int16_t x=pantalla_2.height()-1; x > 25; x-=25)
        pantalla_2.fillRect((pantalla_2.width()-1)/2 -x/2,
(pantalla_2.height()-1)/2 -x/2 , x, x, color);
      break;
    case 3:
      for (int16_t x=pantalla_3.height()-1; x > 25; x-=25)
        pantalla_3.fillRect((pantalla_3.width()-1)/2 -x/2,
(pantalla_3.height()-1)/2 -x/2 , x, x, color);
      break;
    case 4:
      for (int16_t x=pantalla_4.height()-1; x > 25; x-=25)
        pantalla_4.fillRect((pantalla_4.width()-1)/2 -x/2,
(pantalla_4.height()-1)/2 -x/2 , x, x, color);
      break;
    case 5:
      for (int16_t x=pantalla_5.height()-1; x > 25; x-=25)
        pantalla_5.fillRect((pantalla_5.width()-1)/2 -x/2,
(pantalla_5.height()-1)/2 -x/2 , x, x, color);
      break;
    case 6:
      for (int16_t x=pantalla_6.height()-1; x > 25; x-=25)
        pantalla_6.fillRect((pantalla_6.width()-1)/2 -x/2,
(pantalla_6.height()-1)/2 -x/2 , x, x, color);
      break;
    case 7:
      for (int16_t x=pantalla_7.height()-1; x > 25; x-=25)
        pantalla_7.fillRect((pantalla_7.width()-1)/2 -x/2,
(pantalla_7.height()-1)/2 -x/2 , x, x, color);
      break;
    default:
      break;
  }
}

// Hotel
void hotel(){
  pantalla_6.setTextColor(WHITE);
  pantalla_6.setCursor(50,25);
  pantalla_6.setTextSize(8);
  fondo(BLUE,6);
  pantalla_6.print("H");
}

// Ofertas del centro comercial
void ofertas(){
  // howsmall + random() % (howbig - howsmall);
  int aleatorio = random(0, 4);
  switch(aleatorio){
    pantalla_1.setTextColor(BLACK);
    case 1:
      pantalla_1.setCursor(25,30);
      pantalla_1.setTextSize(4);
      fondo(YELLOW,1);
      pantalla_1.println("3x2");
      pantalla_1.setCursor(5,70);
      pantalla_1.setTextSize(2);
```

```
      pantalla_1.print("EN YOGURES");
      break;
    case 2:
      pantalla_1.setCursor(10,25);
      pantalla_1.setTextSize(3);
      fondo(YELLOW,1);
      pantalla_1.println("OFERTA");
      pantalla_1.setCursor(10,60);
      pantalla_1.setTextSize(2);
      pantalla_1.println("NARANJAS");
      pantalla_1.setTextSize(2);
      pantalla_1.setCursor(10,80);
      pantalla_1.println("0,59/kg.");
      break;
    case 3:
      pantalla_1.setCursor(20,15);
      pantalla_1.setTextSize(2);
      fondo(YELLOW,1);
      pantalla_1.println("SEGUNDA");
      pantalla_1.setCursor(25,35);
      pantalla_1.println("UNIDAD");
      pantalla_1.setCursor(12,60);
      pantalla_1.setTextSize(4);
      pantalla_1.println("70%");
      pantalla_1.setTextSize(2);
      pantalla_1.setCursor(85,75);
      pantalla_1.println("dto");
      break;
    default:
      break;
  }
}

// Secuencia de semáforos
void semaforos(int a){
  switch(a){
  case 0:
    semaforo_1(2);
    semaforo_2(2);
    semaforo_3(2);
    semaforo_4(2);
    delay(1500);
    semaforo_1(3);
    semaforo_2(3);
    semaforo_3(3);
    semaforo_4(3);
    delay(1000);
    semaforo_1(1);
    semaforo_4(1);
    break;
  case 1:
    semaforo_1(2);
    semaforo_4(2);
    delay(1500);
    semaforo_1(3);
    semaforo_4(3);
    delay(1500);
```

```
    semaforo_2(1);
    break;
  case 2:
    semaforo_2(2);
    delay(1500);
    semaforo_2(3);
    delay(1500);
    semaforo_3(1);
    semaforo_4(1);
    break;
  case 3:
    semaforo_3(2);
    delay(1500);
    semaforo_3(3);
    delay(1500);
    semaforo_1(1);
    semaforo_4(1);
    break;
  case 4:
    semaforo_1(0);
    semaforo_2(0);
    semaforo_3(0);
    semaforo_4(0);
    delay(800);
    semaforo_1(2);
    semaforo_2(2);
    semaforo_3(2);
    semaforo_4(2);
    delay(800);
    break;
  case 5:
    semaforo_1(2);
    semaforo_2(2);
    semaforo_3(2);
    semaforo_4(2);
    delay(1500);
    semaforo_1(3);
    semaforo_2(3);
    semaforo_3(3);
    semaforo_4(3);
    break;
  default:
    break;
  }
}

void semaforo_1(int color){
  switch(color){
    case 0:
      digitalWrite(2, LOW);
      digitalWrite(3, LOW);
      digitalWrite(4, LOW);
      break;
    case 1:
      digitalWrite(2, HIGH);
      digitalWrite(3, LOW);
      digitalWrite(4, LOW);
```

```
      break;
    case 2:
      digitalWrite(2, LOW);
      digitalWrite(3, HIGH);
      digitalWrite(4, LOW);
      break;
    case 3:
      digitalWrite(2, LOW);
      digitalWrite(3, LOW);
      digitalWrite(4, HIGH);
      break;
    default:
      digitalWrite(2, LOW);
      digitalWrite(3, LOW);
      digitalWrite(4, HIGH);
      break;
    }
}

void semaforo_2(int color){
  switch(color){
     case 0:
      digitalWrite(5, LOW);
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      break;
    case 1:
      digitalWrite(5, HIGH);
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      break;
    case 2:
      digitalWrite(5, LOW);
      digitalWrite(6, HIGH);
      digitalWrite(7, LOW);
      break;
    case 3:
      digitalWrite(5, LOW);
      digitalWrite(6, LOW);
      digitalWrite(7, HIGH);
      break;
    default:
      digitalWrite(5, LOW);
      digitalWrite(6, LOW);
      digitalWrite(7, HIGH);
      break;
    }
}

void semaforo_3(int color){
  switch(color){
    case 0:
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      digitalWrite(10, LOW);
      break;
    case 1:
```

```
      digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
      digitalWrite(10, LOW);
      break;
    case 2:
      digitalWrite(8, LOW);
      digitalWrite(9, HIGH);
      digitalWrite(10, LOW);
      break;
    case 3:
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      digitalWrite(10, HIGH);
      break;
    default:
      digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      digitalWrite(10, HIGH);
      break;
    }
}

void semaforo_4(int color){
  switch(color){
    case 0:
      digitalWrite(11, LOW);
      digitalWrite(12, LOW);
      digitalWrite(13, LOW);
      break;
    case 1:
      digitalWrite(11, HIGH);
      digitalWrite(12, LOW);
      digitalWrite(13, LOW);
      break;
    case 2:
      digitalWrite(11, LOW);
      digitalWrite(12, HIGH);
      digitalWrite(13, LOW);
      break;
    case 3:
      digitalWrite(11, LOW);
      digitalWrite(12, LOW);
      digitalWrite(13, HIGH);
      break;
    default:
      digitalWrite(11, LOW);
      digitalWrite(12, LOW);
      digitalWrite(13, HIGH);
      break;
    }
}
```

8.1.3.- Código fuente ESP32 emisor

```
#include <WiFi.h>

const char* ssid     = “NOMBRE DE LA RED";
const char* password = "CONTRASEÑA";

const char* server = "www.smartblue.es";

const uint16_t port = 80;
const char* host = "192.168.1.1"; // ip o dns

WiFiClient client;

void setup() {
  // Comunicación serie con Arduino MEGA2560
  Serial.begin(115200);

  // Inicia WiFi
  WiFi.begin(ssid, password);

  // Espera a que se conecte a la red WiFi
  while (WiFi.status() != WL_CONNECTED)
    delay(500);

  // Si no se puede conectar al host, termina
  if (!client.connect(host, port))
    return;
}

void loop() {
  // Si se ha recibido algún dato y la conexión es correcta:
  if ((Serial.available() > 0) && (client.connect(server,
port))) {
    // Dato a enviar
    String trama = "";
    // Recibe el estado de los sensores y los almacena en trama
    for(int i=0; i<30; i++)
      trama = trama + (Serial.read() - 48);
    // Construye el campo de datos
    String json = "{\"sensores\":\""+trama+"\"}";
    // Realiza una solicitud HTTP
    client.println("POST /maqueta/api/public/index.php/s HTTP/
1.1");
    client.println("Host: www.smartblue.es");
    client.println("Accept: application/json");
    client.println("Content-Type: application/x-www-form-
urlencoded");
    client.print("Content-Length: ");
    int thisLength = json.length();
    client.println(thisLength);
    client.println("Connection: close");
    client.println();
    client.println(json);
    Serial.println(trama);
  }
}
```

8.1.4.- Código fuente ESP32 receptor

```
#include <WiFi.h>

const char* ssid     = “NOMBRE DE LA RED";
const char* password = “CONTRASEÑA";

const char* server = "www.smartblue.es";

const uint16_t port = 80;
const char * host = "192.168.1.1"; // ip o dns

String msg = "";

// Crea un objeto de tipo WiFiClient
WiFiClient client;

void setup() {
  // Comunicación serie con Arduino MEGA2560
  Serial.begin(115200);

  // Inicia WiFi
  WiFi.begin(ssid, password);

  // Espera a que se conecte a la red WiFi
  while (WiFi.status() != WL_CONNECTED)
    delay(500);

  // Si no se puede conectar al host, termina
  if (!client.connect(host, port))
    return;
}

void loop() {
  if (client.connect(server, port)) {
    // Realiza una solicitud HTTP
    client.println("GET http://www.smartblue.es/maqueta/api/
public/index.php/s? HTTP/1.1");
    client.println("Host: www.smartblue.es");
    client.println("Accept: application/json");
    client.println("Content-Type: application/json");
    client.println("Connection: close");
    client.println();
  }

  delay(500);
  String msg_aux = client.readString();
  msg_aux = msg_aux.substring(msg_aux.length()-32);

  if((msg != msg_aux) && (msg_aux != "")){
    msg = msg_aux;
    Serial.print(msg);
  }
  delay(3000);
}
```

8.1.5.- Código fuente Página Web

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>SmartBlue</title>
  <meta name="description" content="SmartBlue es un sistema de
parking inteligente.">
  <meta name="author" content="Diego Vigil Peláez">
  <!--meta http-equiv="refresh" content="5" -->
  <meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1">

  <link rel="stylesheet" href="/css/bootstrap.min.css" /> <!--
Twitter Bootstrap -->
  <link rel="stylesheet" href="/css/bootstrap-
responsive.min.css" /> <!-- Twitter Bootstrap Responsive -->
  <link rel="stylesheet" href="/css/font-awesome.min.css"> <!--
Font Awesome -->
  <link rel="stylesheet" href="/css/flexslider.css"> <!--
Flexslider -->
  <link rel="stylesheet" href="/css/fancybox/
jquery.fancybox.css"/> <!-- Fancybox -->
  <link rel="stylesheet" href="css/style.css"> <!-- Main
stylesheet -->
  <link rel="stylesheet" href="css/style-responsive.css"><!--
Main stylesheet responsive -->

  <style>
    #canvas {
      display: block;
      padding: 0;
      margin: 0 auto;
    }

    @media(max-width:970px) {
      #canvas {
        width: 545px;
        height: auto;
      }
    }

    @media(max-width:768px) {
      #canvas {
        width: 525px;
        height: auto;
      }
    }

    @media(max-width:480px) {
      #canvas {
        width: 342px;
        height: auto;
      }
    }
  </style>
```

```
  <!--[if IE 7]>
    <link rel="stylesheet" href="css/font-awesome-ie7.min.css">
  <![endif]—>
  <!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/
html5.js"></script>
  <![endif]-->

  <link href='https://fonts.googleapis.com/css?family=Lato|
Open+Sans:400,300' rel='stylesheet' type='text/css'>

  <script type="text/javascript" src="js/jquery.min.js"></
script> <!-- jQuery main file -->
  <script type="text/javascript" src="js/
jquery.flexslider.min.js"></script> <!-- Flexslider -->
  <script type="text/javascript" src="js/
jquery.easing.pack.js"></script> <!-- Easing for Fancybox -->
  <script type="text/javascript" src="js/
jquery.mousewheel.pack.js"></script> <!-- Mousewheel for
Fancybox -->
  <script type="text/javascript" src="js/
jquery.fancybox.pack.js"></script> <!-- Fancybox -->
  <script type="text/javascript" src="js/
jquery.validate.min.js"></script> <!-- Form Validation -->
  <script type="text/javascript" src="js/attraction.js"></
script> <!-- Custom JS -->

  <link rel="shortcut icon" href="img/favicon.ico">
  <link rel="apple-touch-icon" href="img/Logo-
parking-57x57.png">
  <link rel="apple-touch-icon" sizes="72x72" href="img/Logo-
parking-72x72.png">
  <link rel="apple-touch-icon" sizes="114x114" href="img/Logo-
parking-114x114.png">

</head>

<body id="top" class="fixed-nav">
  <section class="header">
    <div class="container">
      <div class="row">
        <div class="span12">
          <div class="logo float-left">
            <a href="/">
              <h1>Smart<span>Blue</span></h1>
            </a>
          </div>

          <ul class="nav hidden-phone hidden-tablet">
            <li><a onclick="registro()">Registro</a></li>
            <li><a href="/">Iniciar sesión</a></li>
          </ul>

          <form name="nav" class="float-right">
            <select class="mobile-nav hidden-desktop" data-
autogenerate="true"></select>
          </form>
```

```
        </div>
      </div>
    </div>
  </section>

  <section class="color-background hero-section">
    <div class="container">
      <div class="row" style="padding-top:70px">
        <div id="menu1" class="span3" align="center"
onclick="menu_1()">
          <option id="parking" type='button' class="button-
dark">Parking</option>
        </div>
        <div id="menu2" class="span3" align="center"
onclick="menu_2()">
          <option id="alumbrado" type='button' class="button-
dark">Alumbrado</option>
        </div>
        <div id="menu3" class="span3" align="center"
onclick="menu_3()">
          <option id="reciclaje" type='button' class="button-
dark">Reciclaje</option>
        </div>
        <div id="menu4" class="span3" align="center"
onclick="menu_4()">
          <option id="semaforos" type='button' class="button-
dark">Semáforos</option>
        </div>
      </div>
    </div>
  </section>

  <section>
    <div class=“container">
      <div class="row">
        <div class="span8" align="center">
          <div id='imagen_1'>
            <img src="/img/Plazas.png" alt="Plazas"></img>
              <canvas id='canvas' width='709px'
height='524px'></canvas>
          </div>
          <div id="imagen_2" style="display:none">
            <img src="/img/Plazas numeradas.png" alt="Zonas"></
img>
          </div>
          <div id="imagen_3" style="display:none">
            <img src="/img/Zonas.png" alt="Plazas"></img>
          </div>
        </div>

        <ul id="parking" class="span4" align="center"
style="margin-top:20px; margin-bottom:20px; display:block">
          <li id="menu1_1" onclick="menu1_1()" class="button-
dark3">Plazas disponibles</li>
          <li id="menu1_2" onclick="menu1_2()" class="button-
dark2">Plazas numeradas</li>
          <li id="menu1_3" onclick="menu1_3()" class="button-
```

```
dark2">Zonas de parking</li>
        </ul>

        <ul id="alumbrado" class="span4" align="center"
style="margin-top:20px; margin-bottom:20px; display:none">
          <li id="menu2_1" onclick="menu2_1()" class="button-
dark2">Mantener encendido</li>
          <li id="menu2_2" onclick="menu2_2()" class="button-
dark2">Mantener apagado</li>
          <li id="menu2_3" onclick="menu2_3()" class="button-
dark3">Funcionamiento normal</li>
        </ul>

        <ul id="reciclaje" class="span4" align="center"
style="margin-top:20px; margin-bottom:20px; display:none">
        <?php
          echo "<input href='/maqueta' class='button-dark2'
id='verde' style='background-color:#333333; cursor:default'
readonly></input>";
          echo "<input href='/maqueta' class='button-dark2'
id='amarillo' style='background-color:#333333; cursor:default'
readonly></input>";
          echo "<input href='/maqueta' class='button-dark2'
id='azul' style='background-color:#333333; cursor:default'
readonly></input>";
        ?>
        </ul>

        <ul id="semaforos" class="span4" align="center"
style="margin-top:20px; margin-bottom:20px; display:none">
          <li id="menu4_1" onclick="menu4_1()" class="button-
dark2">Semáforo 1</li>
          <li id="menu4_2" onclick="menu4_2()" class="button-
dark2">Semáforo 2</li>
          <li id="menu4_3" onclick="menu4_3()" class="button-
dark2">Semáforo 3</li>
          <li id="menu4_4" onclick="menu4_4()" class="button-
dark2">Semáforo 4</li>
          <li id="menu4_5" onclick="menu4_5()" class="button-
dark2">Cortar la calle</li>
          <li id="menu4_6" onclick="menu4_6()" class="button-
dark2">Detener el tráfico</li>
          <li id="menu4_7" onclick="menu4_7()" class="button-
dark3">Funcionamiento normal</li>
        </ul>

      </div>
    </div>
  </section>

  <section class="footer dark-gray-background">
    <div class="container">
      <div class="row">
        <div class="span6">
          <p class="no-margin-bottom"><a href="#">SmartBlue</a>
Project<br>
          Diego Vigil Peláez<br>
```

```
          (+34) 722 386 332<br>
          <a href="#">info@smartblue.es</a></p>
        </div>

        <div class="span6">
          <ul class="social float-right">
            <li><a href="https://plus.google.com"><i
class="icon-google-plus"></i></a></li>
            <li><a href="https://twitter.com"><i class="icon-
twitter"></i></a></li>
            <li><a href="https://www.facebook.com/
SmartBlueES"><i class="icon-facebook"></i></a></li>
          </ul>
          <p class="no-margin-bottom float-right
copyright">Copyright &copy; 2017 <a href="#">SmartBlue</a> Todos
los derechos reservados.</p>
        </div>
      </div>
    </div>
  </section>

</body>

<script type="text/javascript">
  var menu = 1;
  function menu_1(){
    menu = 1;
    document.getElementById("parking").style.display = "block";
    document.getElementById("alumbrado").style.display = "none";
    document.getElementById("reciclaje").style.display = "none";
    document.getElementById("semaforos").style.display = "none";
  }
  function menu_2(){
    menu = 2;
    document.getElementById("parking").style.display = "none";
    document.getElementById("alumbrado").style.display ="block";
    document.getElementById("reciclaje").style.display = "none";
    document.getElementById("semaforos").style.display = "none";
  }
  function menu_3(){
    menu = 3;
    document.getElementById("parking").style.display = "none";
    document.getElementById("alumbrado").style.display = "none";
    document.getElementById("reciclaje").style.display ="block";
    document.getElementById("semaforos").style.display = "none";
  }
  function menu_4(){
    menu = 4;
    document.getElementById("parking").style.display = "none";
    document.getElementById("alumbrado").style.display = "none";
    document.getElementById("reciclaje").style.display = "none";
    document.getElementById("semaforos").style.display ="block";
  }
  function menu1_1(){
    document.getElementById("imagen_1").style.display = "block";
    document.getElementById("imagen_2").style.display = "none";
    document.getElementById("imagen_3").style.display = "none";
```

```
    document.getElementById(“menu1_1").classList.remove("button-
dark2");
    document.getElementById(“menu1_2").classList.remove("button-
dark3");
    document.getElementById(“menu1_3”).classList.remove(“button-
dark3");
    document.getElementById(“menu1_1").classList.add("button-
dark3");
    document.getElementById(“menu1_2").classList.add("button-
dark2");
    document.getElementById(“menu1_3").classList.add("button-
dark2");
  }
  function menu1_2(){
    document.getElementById("imagen_1").style.display = "none";
    document.getElementById("imagen_2").style.display = "block";
    document.getElementById("imagen_3").style.display = "none";
    document.getElementById(“menu1_1").classList.remove("button-
dark3");
    document.getElementById(“menu1_2").classList.remove("button-
dark2");
    document.getElementById(“menu1_3").classList.remove("button-
dark3");
    document.getElementById(“menu1_1").classList.add("button-
dark2");
    document.getElementById(“menu1_2").classList.add("button-
dark3");
    document.getElementById("menu1_3").classList.add("button-
dark2");
  }
  function menu1_3(){
    document.getElementById("imagen_1").style.display = “none";
    document.getElementById("imagen_2").style.display = "none";
    document.getElementById("imagen_3").style.display = "block";
    document.getElementById(“menu1_1").classList.remove("button-
dark3");
    document.getElementById(“menu1_2").classList.remove("button-
dark3");
    document.getElementById(“menu1_3").classList.remove("button-
dark2");
    document.getElementById(“menu1_1").classList.add("button-
dark2");
    document.getElementById(“menu1_2").classList.add("button-
dark2");
    document.getElementById("menu1_3").classList.add("button-
dark3");
  }
  function menu2_1(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open(“POST","api/public/alumbrado.php",true);
    xmlhttp.setRequestHeader("Content-type", “application/x-www-
form-urlencoded");
    xmlhttp.send(“D=1”);
    document.getElementById(“menu2_1”).classList.remove("button-
dark2");
    document.getElementById(“menu2_2").classList.remove("button-
dark3");
```

```
    document.getElementById(“menu2_3”).classList.remove(“button-
dark3");
    document.getElementById(“menu2_1").classList.add("button-
dark3");
    document.getElementById(“menu2_2").classList.add("button-
dark2");
    document.getElementById(“menu2_3").classList.add("button-
dark2");
  }
  function menu2_2(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/alumbrado.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send(“D=2");
    document.getElementById(“menu2_1").classList.remove("button-
dark3");
    document.getElementById(“menu2_2").classList.remove("button-
dark2");
    document.getElementById(“menu2_3").classList.remove("button-
dark3");
    document.getElementById(“menu2_1").classList.add("button-
dark2");
    document.getElementById(“menu2_2").classList.add("button-
dark3");
    document.getElementById(“menu2_3").classList.add("button-
dark2");
  }
  function menu2_3(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/alumbrado.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send(“D=0");
    document.getElementById(“menu2_1").classList.remove("button-
dark3");
    document.getElementById(“menu2_2").classList.remove("button-
dark3");
    document.getElementById(“menu2_3").classList.remove("button-
dark2");
    document.getElementById(“menu2_1").classList.add("button-
dark2");
    document.getElementById(“menu2_2").classList.add("button-
dark2");
    document.getElementById(“menu2_3").classList.add("button-
dark3");
  }
  function menu4_1(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/semaforos.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send(“D=1");
    document.getElementById(“menu4_1").classList.remove("button-
dark2");
    document.getElementById(“menu4_2").classList.remove("button-
dark3");
```

```
   document.getElementById(“menu4_3").classList.remove("button-
dark3");
    document.getElementById(“menu4_4").classList.remove("button-
dark3");
    document.getElementById(“menu4_5").classList.remove("button-
dark3");
    document.getElementById(“menu4_6").classList.remove("button-
dark3");
    document.getElementById(“menu4_7").classList.remove("button-
dark3");
    document.getElementById(“menu4_1").classList.add("button-
dark3");
    document.getElementById(“menu4_2").classList.add("button-
dark2");
    document.getElementById(“menu4_3").classList.add("button-
dark2");
    document.getElementById(“menu4_4").classList.add("button-
dark2");
    document.getElementById(“menu4_5").classList.add("button-
dark2");
    document.getElementById(“menu4_6").classList.add("button-
dark2");
    document.getElementById(“menu4_7").classList.add("button-
dark2");
    setTimeout('menu4_7()', 12000);
  }
  function menu4_2(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/semaforos.php",true);
    xmlhttp.setRequestHeader("Content-type", “application/x-www-
form-urlencoded”);
    xmlhttp.send(“D=2");
    document.getElementById(“menu4_1").classList.remove("button-
dark3");
    document.getElementById(“menu4_2").classList.remove("button-
dark2");
    document.getElementById(“menu4_3").classList.remove("button-
dark3");
    document.getElementById(“menu4_4").classList.remove("button-
dark3");
    document.getElementById(“menu4_5").classList.remove("button-
dark3");
    document.getElementById(“menu4_6").classList.remove("button-
dark3");
    document.getElementById(“menu4_7").classList.remove("button-
dark3");
    document.getElementById(“menu4_1").classList.add("button-
dark2");
    document.getElementById(“menu4_2").classList.add("button-
dark3");
    document.getElementById(“menu4_3”).classList.add("button-
dark2");
    document.getElementById(“menu4_4").classList.add("button-
dark2");
    document.getElementById(“menu4_5”).classList.add("button-
dark2");
    document.getElementById(“menu4_6").classList.add("button-
```

```
dark2");
    document.getElementById("menu4_7").classList.add("button-
dark2");
    setTimeout('menu4_7()', 12000);
  }
  function menu4_3(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/semaforos.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send("D=3");
    document.getElementById("menu4_1").classList.remove("button-
dark3");
    document.getElementById("menu4_2").classList.remove("button-
dark3");
    document.getElementById("menu4_3").classList.remove("button-
dark2");
    document.getElementById("menu4_4").classList.remove("button-
dark3");
    document.getElementById("menu4_5").classList.remove("button-
dark3");
    document.getElementById("menu4_6").classList.remove("button-
dark3");
    document.getElementById("menu4_7").classList.remove("button-
dark3");
    document.getElementById("menu4_1").classList.add("button-
dark2");
    document.getElementById("menu4_2").classList.add("button-
dark2");
    document.getElementById("menu4_3").classList.add("button-
dark3");
    document.getElementById("menu4_4").classList.add("button-
dark2");
    document.getElementById("menu4_5").classList.add("button-
dark2");
    document.getElementById("menu4_6").classList.add("button-
dark2");
    document.getElementById("menu4_7").classList.add("button-
dark2");
    setTimeout('menu4_7()', 12000);
  }
  function menu4_4(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/semaforos.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send("D=4");
    document.getElementById("menu4_1").classList.remove("button-
dark3");
    document.getElementById("menu4_2").classList.remove("button-
dark3");
    document.getElementById("menu4_3").classList.remove("button-
dark3");
    document.getElementById("menu4_4").classList.remove("button-
dark2");
    document.getElementById("menu4_5").classList.remove("button-
dark3");
```

```
    document.getElementById(“menu4_6").classList.remove("button-
dark3");
    document.getElementById(“menu4_7").classList.remove("button-
dark3");
    document.getElementById(“menu4_1").classList.add("button-
dark2");
    document.getElementById(“menu4_2").classList.add("button-
dark2");
    document.getElementById(“menu4_3").classList.add("button-
dark2");
    document.getElementById(“menu4_4”).classList.add("button-
dark3");
    document.getElementById(“menu4_5").classList.add("button-
dark2");
    document.getElementById(“menu4_6").classList.add("button-
dark2");
    document.getElementById("menu4_7").classList.add("button-
dark2");
    setTimeout('menu4_7()', 12000);
  }
  function menu4_5(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/semaforos.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send(“D=5");
    document.getElementById(“menu4_1").classList.remove("button-
dark3");
    document.getElementById(“menu4_2").classList.remove("button-
dark3");
    document.getElementById(“menu4_3").classList.remove("button-
dark3");
    document.getElementById(“menu4_4").classList.remove("button-
dark3");
    document.getElementById(“menu4_5").classList.remove("button-
dark2");
    document.getElementById(“menu4_6").classList.remove("button-
dark3");
    document.getElementById(“menu4_7").classList.remove("button-
dark3");
    document.getElementById(“menu4_1").classList.add("button-
dark2");
    document.getElementById(“menu4_2").classList.add("button-
dark2");
    document.getElementById(“menu4_3").classList.add("button-
dark2");
    document.getElementById(“menu4_4”).classList.add("button-
dark2");
    document.getElementById(“menu4_5").classList.add("button-
dark3");
    document.getElementById(“menu4_6").classList.add("button-
dark2");
    document.getElementById("menu4_7").classList.add("button-
dark2");
  }
  function menu4_6(){
    var xmlhttp = new XMLHttpRequest();
```

```
    xmlhttp.open("POST","api/public/semaforos.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send("D=6");
    document.getElementById("menu4_1").classList.remove("button-
dark3");
    document.getElementById("menu4_2").classList.remove("button-
dark3");
    document.getElementById("menu4_3").classList.remove("button-
dark3");
    document.getElementById("menu4_4").classList.remove("button-
dark3");
    document.getElementById("menu4_5").classList.remove("button-
dark3");
    document.getElementById("menu4_6").classList.remove("button-
dark2");
    document.getElementById("menu4_7").classList.remove("button-
dark3");
    document.getElementById("menu4_1").classList.add("button-
dark2");
    document.getElementById("menu4_2").classList.add("button-
dark2");
    document.getElementById("menu4_3").classList.add("button-
dark2");
    document.getElementById("menu4_4").classList.add("button-
dark2");
    document.getElementById("menu4_5").classList.add("button-
dark2");
    document.getElementById("menu4_6").classList.add("button-
dark3");
    document.getElementById("menu4_7").classList.add("button-
dark2");
  }
  function menu4_7(){
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST","api/public/semaforos.php",true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlhttp.send("D=0");
    document.getElementById("menu4_1").classList.remove("button-
dark3");
    document.getElementById("menu4_2").classList.remove("button-
dark3");
    document.getElementById("menu4_3").classList.remove("button-
dark3");
    document.getElementById("menu4_4").classList.remove("button-
dark3");
    document.getElementById("menu4_5").classList.remove("button-
dark3");
    document.getElementById("menu4_6").classList.remove("button-
dark3");
    document.getElementById("menu4_7").classList.remove("button-
dark2");
    document.getElementById("menu4_1").classList.add("button-
dark2");
    document.getElementById("menu4_2").classList.add("button-
dark2");
```

```
    document.getElementById(“menu4_3").classList.add("button-
dark2");
    document.getElementById(“menu4_4").classList.add("button-
dark2");
    document.getElementById(“menu4_5").classList.add("button-
dark2");
    document.getElementById(“menu4_6").classList.add("button-
dark2");
    document.getElementById("menu4_7").classList.add("button-
dark3");
  }
  </script>

  <script type="text/javascript">
    var color;
    var enviado1, enviado2, enviado3;
    $(document).ready(function() {
      function actualizarMaqueta(){
        var datos;
        $(function(){
          $.ajax({
            url: "api/public/secuencia.txt",
            async: false,
            cache: false,
            dataType: “text",
            success: function( data, textStatus, jqXHR ) {
              datos = data.trim();
            }
          });
        });

        var img = document.getElementById("imagen_1");
        var cnvs = document.getElementById("canvas");
        cnvs.style.position = "absolute";
        cnvs.style.left = img.offsetLeft + "px";
        cnvs.style.top = img.offsetTop + "px";
        var ctx = cnvs.getContext(“2d");

        function dibujar(a, b, c, d){
          ctx.beginPath();
          ctx.arc(a, b, c, 0, 2 * Math.PI, false);
          if(datos.charAt(d) == '0'){
            ctx.fillStyle = '#ff0000';
          }else{
            ctx.fillStyle = '#00ff00';
          }
          ctx.fill();
          ctx.closePath();
        }

        dibujar(140, 393, 10, 0);
        dibujar(139, 418, 10, 1);
        dibujar(65, 347, 10, 2);
        dibujar(62, 370, 10, 3);
        dibujar(59, 393, 10, 4);
        dibujar(56, 417, 10, 5);
        dibujar(53, 442, 10, 6);
```

```
        dibujar(315, 395, 10, 7);
        dibujar(315, 420, 10, 8);
        dibujar(230, 395, 10, 9);
        dibujar(230, 420, 10, 10);
        dibujar(442, 368, 10, 11);
        dibujar(441, 342, 10, 12);
        dibujar(435, 261, 10, 13);
        dibujar(434, 238, 10, 14);
        dibujar(432, 215, 10, 15);
        dibujar(580, 436, 10, 16);
        dibujar(575, 387, 10, 17);
        dibujar(568, 340, 10, 18);
        dibujar(558, 263, 10, 19);
        dibujar(550, 220, 10, 20);
        dibujar(513, 68, 8, 21);
        dibujar(515, 87, 8, 22);
        dibujar(517, 106, 8, 23);
        dibujar(118, 138, 10, 24);
        dibujar(74, 290, 10, 25);

        if(menu == 3){
          if(datos.charAt(26) == 0){
            document.getElementById("verde").value = "Verde:
VACÍO”;
            enviado1 = false;
          }else{
            document.getElementById("verde").value = "Verde:
LLENO”;
            if(!enviado1){
              correo("verde");
              enviado1 = true;
            }
          }
          if(datos.charAt(27) == 0){
            document.getElementById("amarillo").value =
"Amarillo: VACÍO”;
            enviado2 = false;
          }else{
            document.getElementById("amarillo").value =
"Amarillo: LLENO”;
            if(!enviado2){
              correo(“amarillo");
              enviado2 = true;
            }
          }
          if(datos.charAt(28) == 0){
            document.getElementById("azul").value = "Azul:
VACÍO”;
            enviado3 = false;
          }else{
            document.getElementById("azul").value = "Azul:
LLENO”;
            if(!enviado3){
              correo(“azul");
              enviado3 = true;
            }
          }
```

```
        }
      }
      setInterval(function(){ actualizarMaqueta(); }, 1000);
    });
    function correo(color){
      var xmlhttp = new XMLHttpRequest();
      xmlhttp.open("POST", "correo/mail.php", true);
      xmlhttp.setRequestHeader("Content-type", "application/x-
www-form-urlencoded");
      xmlhttp.send("color=" + color);
    }
    function registro(){
      alert("Actualmente no se permite ningún registro.");
    }
  </script>
</html>
```

8.1.6.- Código fuente App Móvil

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="es.smartblue.smartblue" >

   <uses-permission android:name="android.permission.INTERNET"/>

   <application
      android:allowBackup="true"
      android:icon="@drawable/icon"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/AppTheme">
      <activity
         android:name=".FullscreenActivity"
         android:configChanges="orientation|keyboardHidden|screenSize"
         android:label="@string/app_name"
         android:theme="@style/FullscreenTheme"
         android:screenOrientation="landscape" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER"/>
         </intent-filter>
      </activity>
  </application>

</manifest>
```

FullscreenActivity.java

```
package es.smartblue.smartblue;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class FullscreenActivity extends AppCompatActivity {

   private WebView webview ;
   @Override

   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_fullscreen);

      //webview use to call own site
      webview = (WebView)findViewById(R.id.webView);

      webview.setWebViewClient(new WebViewClient());
      webview.getSettings().setJavaScriptEnabled(true);
      webview.getSettings().setDomStorageEnabled(true);
      webview.loadUrl("http://www.smartblue.es/app");
   }
}
```

8.2.- Encuesta

8.2.1.- Preguntas realizadas

1- ¿Cómo es tu experiencia en una ciudad a la hora de buscar parking?

……..

……..

……..

2- ¿A qué horas o días sueles buscar un parking?

……..

3- ¿Te gustaría tener algún sistema que te ayudara a encontrar parking más fácilmente?

……..

4- ¿Cuál crees que sería la forma más sencilla de tener un sistema como ese? ¿Cómo lo harías?

……..

……..

……..

5a- ¿Has oído hablar de algún sistema que proporcione ya la solución a este problema? ¿Cuál?

……..

5b- En caso afirmativo, ¿lo has usado alguna vez? ¿Por qué?

……..

……..

6- ¿Dónde prefieres estacionar tu vehículo fuera de casa? (En la zona azul, un parking privado, un parking público, gratis en la calle...)

……..

7- ¿Sueles consumir un producto porque te lo hayan aconsejado? …….

8- ¿Te guías por las redes sociales para consumir un producto? …….

9- ¿Utilizarías el parking de un hotel si funcionara del mismo modo que un parking público normal?

……..

10- ¿Tienes smartphone? …… ¿Tienes tarifa de internet? ……..

11- ¿Te resulta fácil usar las aplicaciones en general? …….

12- ¿Cuál es la prioridad que le das al funcionamiento de una aplicación móvil? (Que sea clara, rápida, sencilla...)

……..

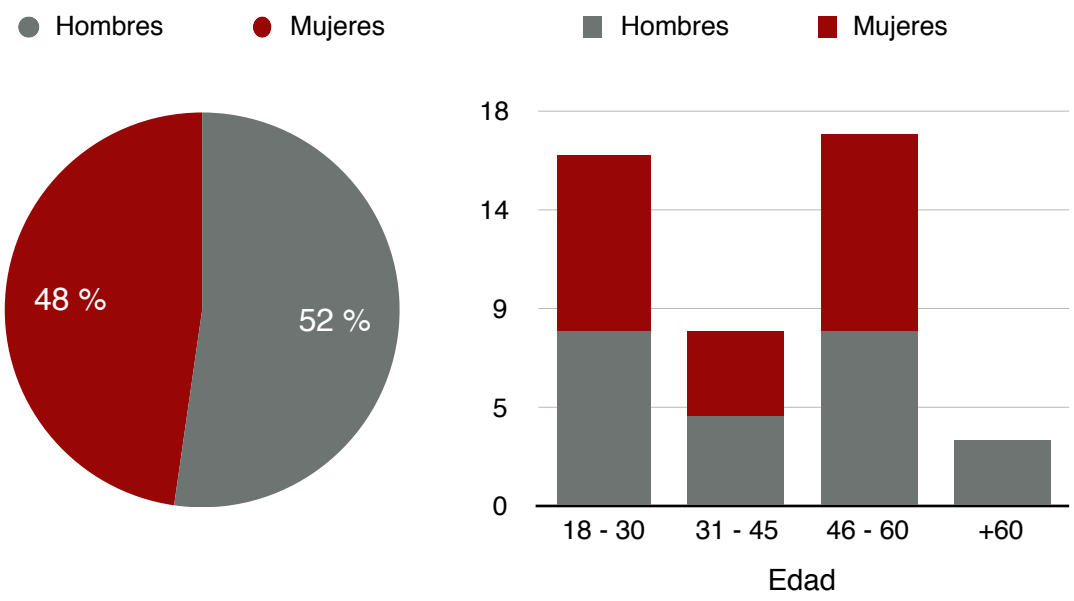
8.2.2.- Criterios de validación

Para confirmar o descartar cada una de las hipótesis, se han establecido los siguientes criterios de validación:

- Las hipótesis serán válidas si superan el 70% de respuestas afirmativas.
- Las hipótesis serán inválidas si no superan el 30% de respuestas afirmativas.
- Las hipótesis continuarán en proceso de validación si se obtiene un resultado de entre el 30 y el 70 por ciento.

8.2.3.- Métrica

Se han realizado un total de 44 encuestas con los resultados que se muestran a continuación:



Hombres	18	20	21	22	27	30	30	30	32	33	40	45
Mujeres	19	20	21	22	25	27	28	30	31	40	41	45

Hombres	50	50	50	50	53	54	56	57	61	63	70	
Mujeres	47	48	52	52	55	56	56	60	60			

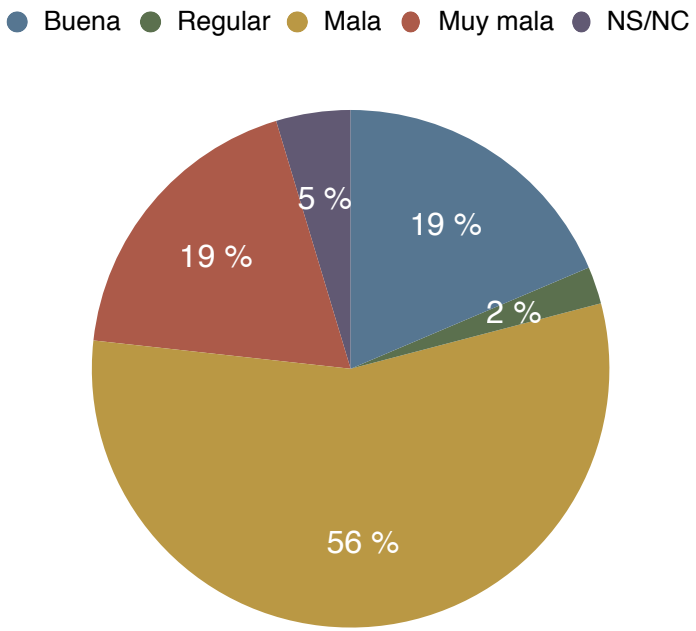
Tabla 16: Edades de los encuestados por género

Primera hipótesis:

- Creo que mi cliente tiene un problema a la hora de aparcar.

Para ello se incluye la siguiente pregunta en el cuestionario:

1- ¿Cómo es tu experiencia en una ciudad a la hora de buscar parking?



Tres de cada cuatro encuestados manifiestan una mala o muy mala experiencia a la hora de estacionar su vehículo en una ciudad.

- "*A veces de forma caótica y hasta agotar paciencia dando vueltas y vueltas.*"
- *"Pérdida de tiempo y combustible."*
- *"A veces imposible o pagando precios abusivos".*
- *"Muy mala. En la calle es casi imposible".*
- *"En la zona blanca bastante difícil."*
- *"En una ciudad grande es muy difícil aparcar en la calle."*
- *"Terrible."*
- *"Nefasto. Normalmente doy vueltas sin cesar."*
- *"Me cuesta mucho aparcar."*
- *"Suelo dar muchas vueltas."*
- *"Estresante."*
- *"Siempre doy muchas vueltas."*
- *…*

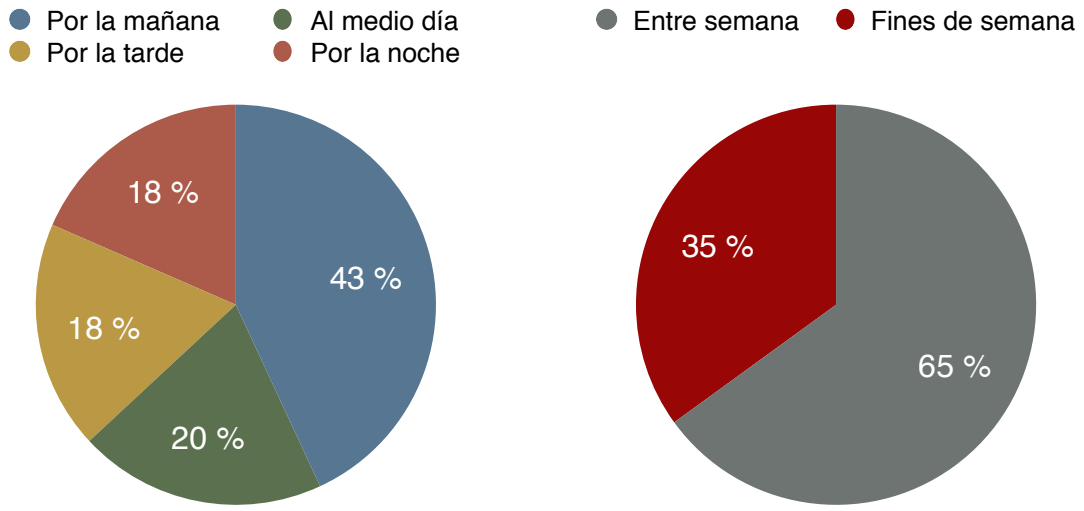
Por tanto, la hipótesis se considera **VALIDADA**.

Segunda hipótesis:

- Creo que mi usuario utilizará mi servicio entre las 8 y las 10 de la mañana y las 6 y las 8 de la tarde.

Para ello se incluye la siguiente pregunta en el cuestionario:

2- ¿A qué horas o días sueles buscar un parking?



Dos de cada cinco encuestados necesita estacionar su vehículo en horario de mañanas frente a cualquier otra franja del día, siendo más necesario entre semana.

- "*Por las mañanas para ir a clase.*"
- *"Horas de comer o cenar."*
- *"Los fines de semana por la tarde".*
- *"Medio día. Noche".*
- *"Al medio día."*
- *"Por la mañana en días laborables o en las noches del fin de semana."*
- *"Cuatro días a las tres de la tarde."*
- *"Normalmente a primera hora de horario comercial."*
- *"Por semana a primera hora."*
- *"En días laborables."*
- *"Fin de semana. Por las tardes."*
- *"Por las mañanas."*
- *…*

Por tanto, la hipótesis se considera **EN PROCESO DE VALIDACIÓN**.

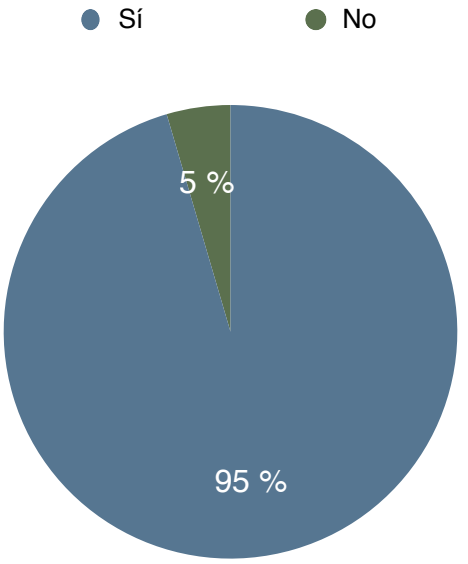
Se reconsidera la hipótesis pudiendo ser un servicio necesario en horas variables, tanto a horas puntas como en fines de semana, aunque puede considerarse un repunte en horario matutino laboral.

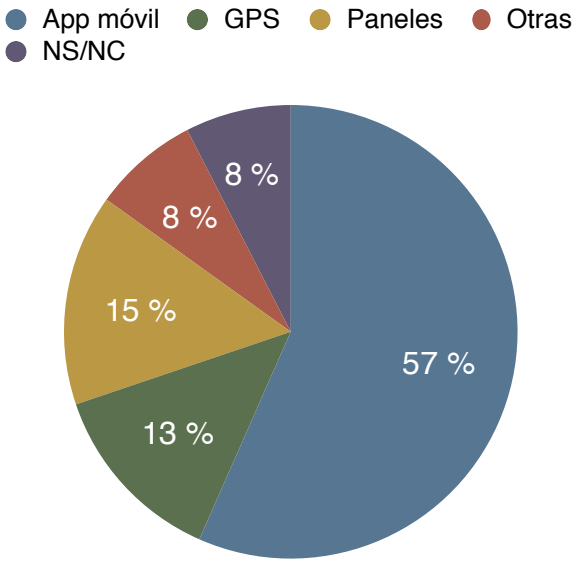
Tercera hipótesis:

- Creo que este problema puede ser resuelto mediante la implementación de una app móvil.

Para ello se incluyen las siguientes preguntas en el cuestionario:

3- ¿Te gustaría tener algún sistema que te ayudara a encontrar parking más fácilmente?
4- ¿Cuál crees que sería la forma más sencilla de tener un sistema como ese? ¿Cómo lo harías?





La inmensa mayoría de los encuestados desearía tener un sistema que les ayude a encontrar parking optando por una aplicación móvil como la mejor solución, aunque debería complementarse con paneles informativos en las calles o geolocalización.

- "*Facilitaría mucho a la gente con prisa.*"
- *"Sí, por supuesto."*
- *"Sería bueno".*
- *"Sí, algo que diga dónde hay sitios libres".*

- *"En el teléfono móvil. Introduciendo el nombre de la ciudad o localidad y que nos indicase de la misma manera que el navegador."*
- *"Con una aplicación móvil que permitiera localizar y saber en tiempo real por zonas urbanas: las plazas vacantes, sensores, detectores, antenas, coche-móvil-zonas permitidas, etc…"*
- *"Una aplicación de smartphone."*
- *"Con una app para el móvil."*
- *"Una aplicación en el móvil conectada al coche."*
- *"Localización de plazas por GPS."*
- *"En la calle con carteles digitales con el porcentaje de sitios libres."*

- *“Un aparcamiento público en el centro de las ciudades para ayudar al comercio local.”*
- *“Un sistema que mande un mensaje avisando donde hay un sitio libre a través del parquímetro según los tickets que emita.”*
- *“Una app como todo.”*
- *“Con aplicaciones.”*
- *“Por aplicaciones móviles.”*
- *“Habilitando más zonas para aparcar.”*
- *“Una app con geolocalización.”*
- *“Una app estaría bien.”*
- *…*

Con una aplicación móvil sería **insuficiente** para poder validar esta hipótesis al no alcanzar el mínimo del 70% establecido en los criterios de validación, por lo que se reconsidera complementándolo con paneles indicativos en las calles y la posibilidad de incluir el sistema de navegación por GPS.

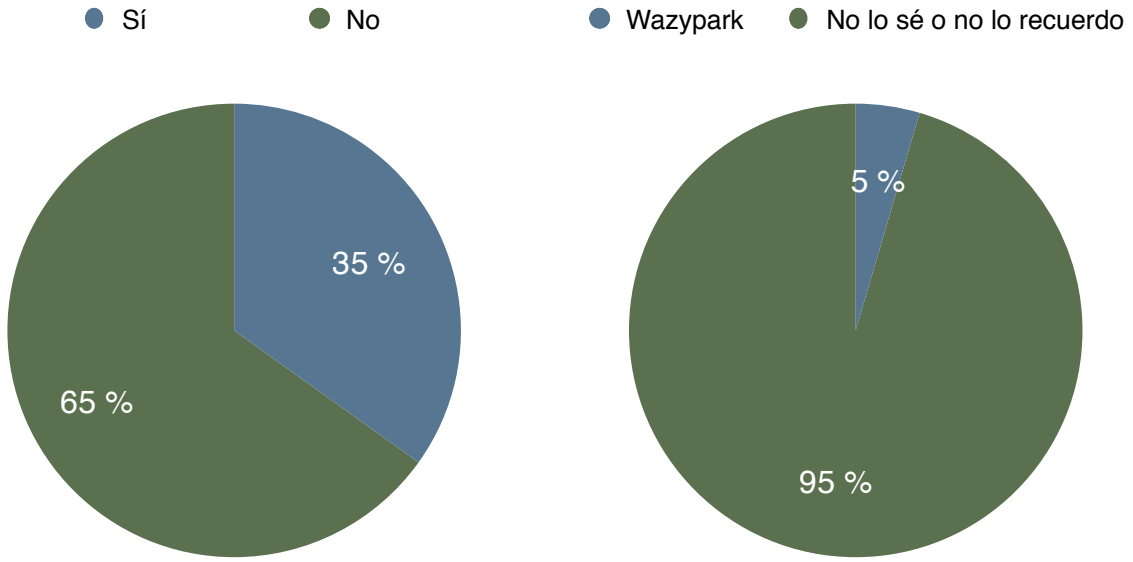
Teniendo en cuenta lo anterior, se puede considerar **VALIDADA**.

Cuarta hipótesis:

- Creo que mi principal competidor en el mercado será *Wazypark*.

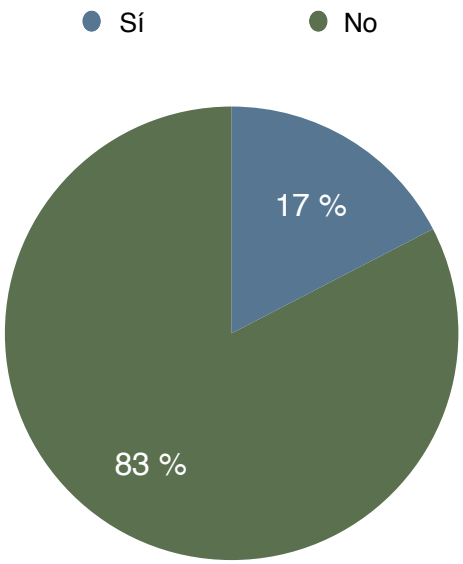
Para ello se incluyen las siguientes preguntas en el cuestionario:

5a- ¿Has oído hablar de algún sistema que proporcione ya la solución a este problema? ¿Cuál?



Dos de cada tres encuestados desconoce la existencia de este servicio y a penas un cinco por ciento recuerda la marca de la competencia.

5b- En caso afirmativo, ¿lo has usado alguna vez? ¿Por qué?



- *"Sí, pero muchas veces el parking no está libre."*
- *"No, nunca la he utilizado."*
- *"He oído hablar de ella pero no la he usado."*
- *"No, no me la he encontrado."*
- *"No, en la zona no están en práctica."*
- *…*

Ocho de cada diez personas que conocen la aplicación de la competencia, nunca ha utilizado su servicio.

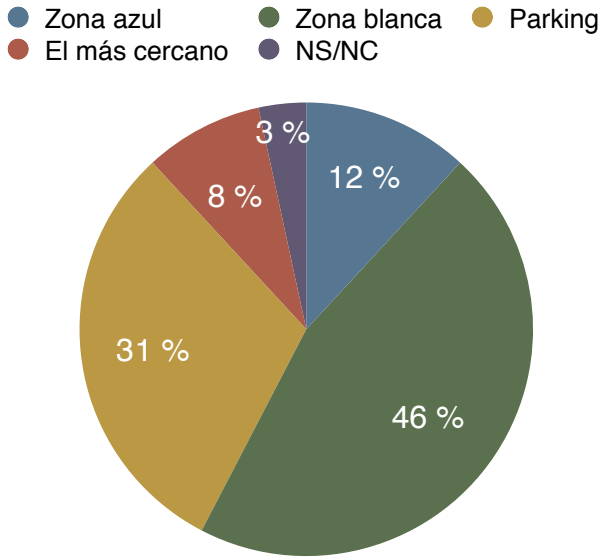
De estas preguntas se puede concluir que la competencia puede ser fácilmente absorbible al pasar totalmente desapercibida para la mayoría de los encuestados. Por tanto, la hipótesis se considera **INVALIDADA**.

Quinta hipótesis:

- Creo que los distintos tipos de cliente que tengo son los ayuntamientos, hoteles, centros comerciales y usuarios privados.

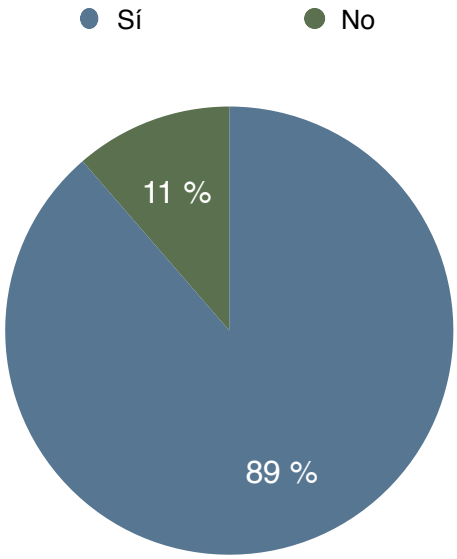
Para ello se incluyen las siguientes preguntas en el cuestionario:

6- ¿Dónde prefieres estacionar tu vehículo fuera de casa? (En la zona azul, un parking privado, un parking público, gratis en la calle…)



Más de la mitad de los encuestados está dispuesto a pagar por estacionar su vehículo en diferentes ubicaciones.

9- ¿Utilizarías el parking de un hotel si funcionara del mismo modo que un parking público normal?



- *"Sí, si su precio fuese el mismo."*
- *"Sí, interesante."*
- *…*

Nueve de cada diez encuestados estarían dispuestos a pagar por estacionar en las plazas libres de los hoteles, convirtiéndoles en potenciales clientes y considerando parte de la hipótesis como **VALIDADA**.

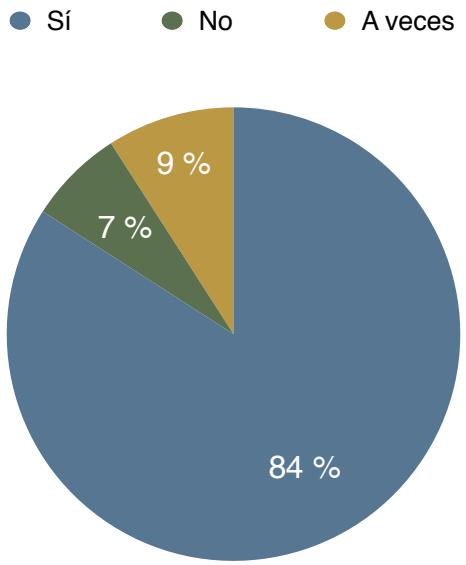
Sin embargo, y en base a los datos de los que se dispone, el resto de esta hipótesis deberá continuar aun **EN PROCESO DE VALIDACIÓN**.

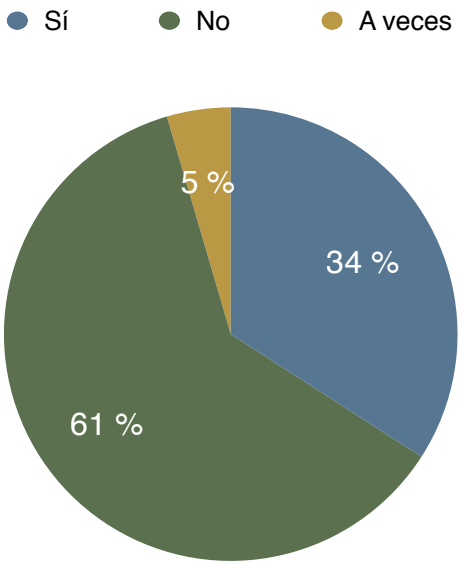
Sexta hipótesis:

- Creo que conseguiré la mayor parte de mis clientes a través del boca a boca y las redes sociales.

Para ello se incluyen las siguientes preguntas en el cuestionario:

7- ¿Sueles consumir un producto porque te lo hayan aconsejado?
8- ¿Te guías por las redes sociales para consumir un producto?





La gran mayoría de los encuestados consumiría un producto gracias al boca a boca, mientras que a penas uno de cada tres atendería a las redes sociales.

- *"Si me lo aconseja un amigo, sí."*
- *"Sí, el boca a boca es importante."*

- *"Generalmente no me guío por las redes sociales para consumir un producto".*
- *"Depende del producto."*
- *…*

Por tanto, se puede concluir que el boca a boca es fundamental para conseguir adeptos a este servicio, haciendo que la parte de la hipótesis que hace referencia a este punto se considere **VALIDADA**.

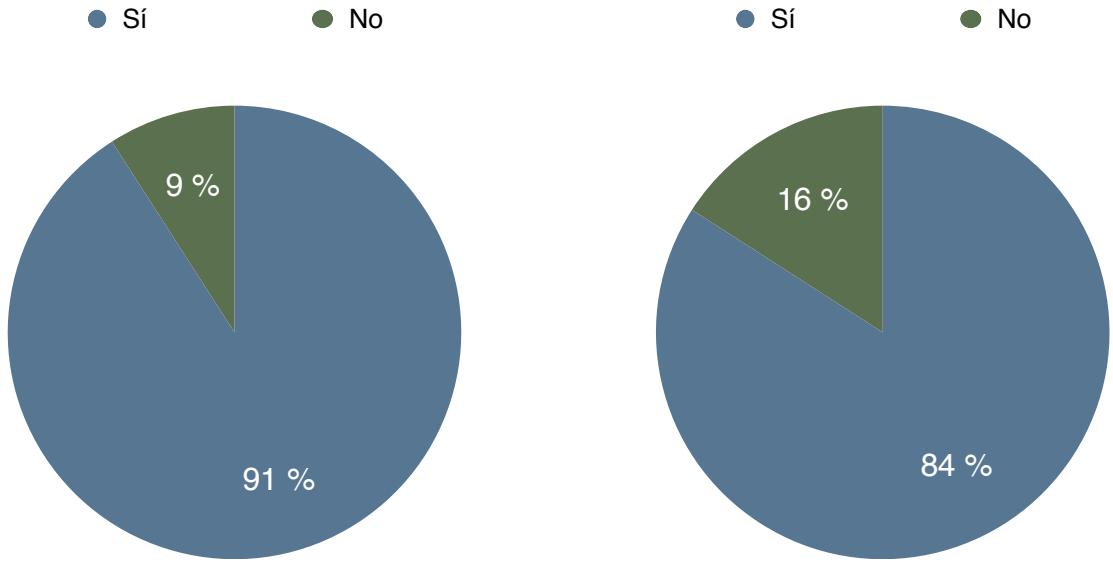
Mientras que las redes sociales no parecen tener mucho poder a la hora de captar nuevos clientes, aunque según estos datos no es un valor despreciable. De este modo, la parte de la hipótesis que hace referencia a la promoción en redes sociales se mantiene **EN PROCESO DE VALIDACIÓN**.

Séptima hipótesis:

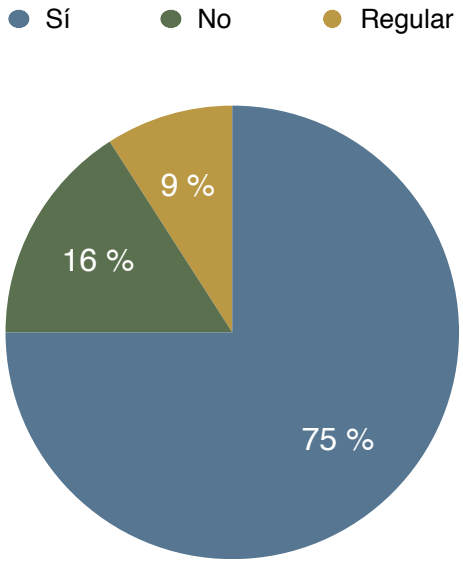
- Creo que sus principales barreras son las tecnológicas, ya que la solución que se propone es a través de un *smartphone* utilizando internet.

Para ello se incluyen las siguientes preguntas en el cuestionario:

10- ¿Tienes *smartphone*? ¿Tienes tarifa de internet?



11- ¿Te resulta fácil usar las aplicaciones en general?



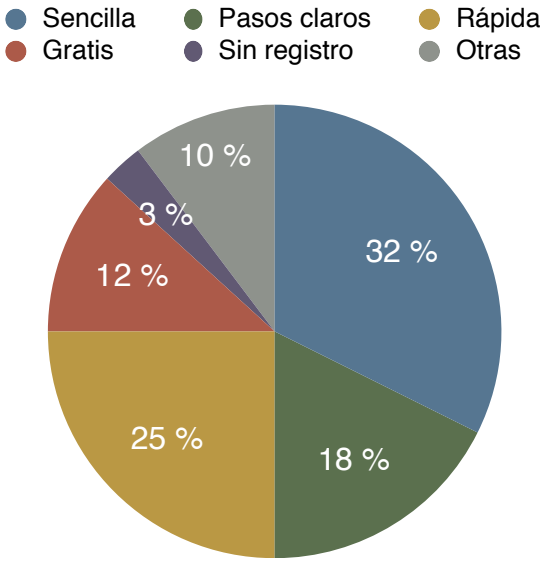
Es posible que la tecnología móvil pueda presentar algún inconveniente en aquellos que no estén muy familiarizados, pero la gran mayoría no parece presentar inconvenientes. Además, el 84% dispone de tarifa de internet en el móvil, condición indispensable para poder utilizar el servicio. Con lo cual, esta hipótesis se puede considerar **INVALIDADA**.

Octava hipótesis:

- Creo que mi mayor riesgo en el servicio es que parezca demasiado complejo, caro y poco eficiente.

Para ello se incluye la siguiente pregunta en el cuestionario:

12- ¿Cuál es la prioridad que le das al funcionamiento de una aplicación móvil? (Que sea clara, rápida, sencilla…)



- *"Principalmente que sea rápida y clara. De no ser así, no la utilizo."*
- *"Que sea fácil, legible y que se entienda fácilmente la mecánica de la aplicación."*
- *"Que no moleste mucho."*
- *"Que sea gratis y rápida."*
- *"Que sea gratis y sin inscripciones."*
- *"Con los pasos muy claros."*
- *"Con letras claras y no muchos pasos."*
- *"Que sea muy interactiva."*
- *"Que sea fácil, rápida y sin anuncios."*
- *"Clara, sin muchas opciones."*
- *"Rápida y eficaz."*
- *"Fácil de usar."*
- *"Todas ellas."*
- *…*

Son muchos los requisitos de los posibles usuarios, y cumplirlos todos para satisfacer a los consumidores será un factor muy importante a tener en cuenta que puede poner en riesgo la implantación de este servicio. Por lo tanto, esta hipótesis se considera **VALIDADA**.